

TATIANA ANNONI PAZETO

**“ESCALONADORES DE TRÁFEGO PARA GARANTIR
QOS EM REDES CONVERGENTES E CORPORAIS”**

***“TRAFFIC SCHEDULERS TO GUARANTEE THE QOS IN
CONVERGENT AND BODY NETWORKS”***

**CAMPINAS
2012**

**UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO**

TATIANA ANNONI PAZETO

**“ESCALONADORES DE TRÁFEGO PARA GARANTIR QOS EM
REDES CONVERGENTES E CORPORAIS”**

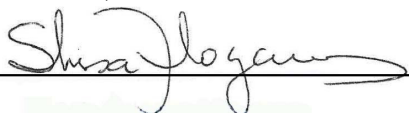
***“TRAFFIC SCHEDULERS TO GUARANTEE THE QOS IN
CONVERGENT AND BODY NETWORKS”***

Tese de doutorado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas para obtenção do título de Doutora em Engenharia Elétrica, na área de Telecomunicações e Telemática.

*Doctorate thesis presented to the Electrical Engineering
Postgraduation Programm of the School of Engineering Electrical of the University of
Campinas to obtain the Ph.D. grade in Engineering Electrical, in field of
Telecommunications and Telematics.*

Orientador: Prof. Dr. Shusaburo Motoyama
Tutor: Associate Professor Shusaburo Motoyama

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL DA TESE
DEFENDIDA PELO ALUNO, E ORIENTADA PELO PROF. DR.



**CAMPINAS
2012**

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

P298e Pazeto, Tatiana Annoni
Escalonadores de tráfego para garantir QOS em redes convergentes e corporais / Tatiana Annoni Pazeto. -- Campinas, SP: [s.n.], 2012.

Orientador: Shusaburo Motoyama.
Tese de Doutorado - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Redes de computadores. I. Motoyama, Shusaburo, 1944-. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Título em Inglês: Traffic schedulers to guarantee the QOS in convergent and body networks

Palavras-chave em Inglês: Computer networks

Área de concentração: Telecomunicações e Telemática

Titulação: Doutora em Engenharia Elétrica

Banca examinadora: Elizabeth Sueli Specialski, Magda Patrícia Caldeira Arantes, Akebo Yamakami, Maurício Ferreira Magalhães

Data da defesa: 13-07-2012

Programa de Pós Graduação: Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE DOUTORADO

Candidata: Tatiana Annoni Pazeto

Data da Defesa: 13 de julho de 2012

Título da Tese: "Escalonadores de Tráfego para Garantir QOS em Redes Convergentes e Corporais "

Prof. Dr. Shusaburo Motoyama (Presidente):

Shusaburo Motoyama

Profa. Dra. Elizabeth Sueli Specialski:

Elizabeth Sueli Specialski

Profa. Dra. Magda Patrícia Caldeira Arantes:

Magda Patrícia Caldeira Arantes

Prof. Dr. Akebo Yamakami:

Akebo Yamakami

Prof. Dr. Maurício Ferreira Magalhães:

Maurício Ferreira Magalhães

Dedico esta obra a pessoa mais ilustre e interessada nisso: EU

Para que em todos os momentos que eu pensar em desistir de algo, possa recordar das experiências vivenciadas ao longo da trajetória do Doutorado, e do gostinho de vitória por ter superado tudo e concluído este sonho ou pesadelo, sei lá.

Agradecimentos

Agradeço a Deus por ter me feito uma pessoa com garra, perseverança e tenacidade, de tal forma que hoje percebo a vitória e superação.

A meus pais, pela paciência, compreensão e apoio ao longo destes anos de jornada, os quais serão super importantes para a escolha profissional feita.

A meus irmãos e respectivas famílias, por entender que minha ausência muitas vezes era necessária para que eu pudesse cumprir e chegar ao objetivo almejado.

Ao meu orientador, Prof. Shusaburo Motoyama, a quem admiro e respeito por saber os momentos em que há necessidade de ser exigente e doce, mas sobretudo, humano.

Aos meus orientandos, com quem convivi muito nos últimos anos, trabalhando, debatendo e vivendo o dia a dia. Em especial, a Renato Moraes Silva, que indiretamente tem muita contribuição para que eu concluísse esta etapa. Afinal, foram horas de programação e testes de mesa dos programas.

A todos os amigos, que de algum modo contribuíram para esta vitória.

Agradeço também ao CNPq por ter financiado o presente trabalho.

Resumo

As redes atuais, baseadas em tecnologia IP, transportam uma variedade de tráfegos, tais como voz, dados e vídeos (tráfego multimídia) e são denominadas de redes convergentes. Outras redes estão em desenvolvimento para aplicações específicas como Redes de Sensores Corporais Sem Fio (RSC).

O problema de prover a qualidade de serviço (QoS – Quality of Service) de cada tipo de tráfego em redes convergentes e em RSC, é essencial, pois os tráfegos exigem diferentes requisitos de qualidade. Um dos principais parâmetros para prover QoS nessas redes é o escalonador de tráfegos. O objetivo principal desta tese é analisar e propor escalonadores de tráfego para a rede convergente e para RSC.

O escalonador de tráfegos proporciona uma utilização mais equitativa da banda disponível. Como a rede baseada em IP foi originalmente projetada para transportar somente tráfegos de dados, é estudada, nesta tese, a influência do tráfego multimídia no desempenho e no projeto do escalonador. O estudo é realizado através do desenvolvimento de várias plataformas de simulações que contêm os vários tipos de tráfegos, um buffer de armazenamento de pacotes, um link de saída e os vários tipos de escalonamento.

Como o escalonador FIFO foi e continua sendo o mais utilizado, esse escalonador foi tomado como referencia para confrontar os resultados obtidos com o escalonador DRR e com o outro modelo inédito de escalonamento que usa em seu cálculo de distribuição de quotas, o conceito de banda efetiva. O escalonador DRR e o de banda efetiva podem fazer

distinção entre os tráfegos, de modo que a alocação de bandas entre os vários tráfegos se torne mais justa e atenda aos requisitos de QoS.

Pelos resultados obtidos, pode-se constatar que a solução de escalonamento proposta consegue controlar a perda de pacotes, mas o atraso deve ser melhor investigado. Além disso, o escalonador DRR é mais indicado para tráfego multimídia se comparado ao escalonador FIFO.

Na maioria das propostas examinadas na literatura, o escalonador mais utilizado para RSC é aquele baseado em técnica TDM. Existem poucos estudos em que o escalonador é baseado em serviço cíclico, conhecido na literatura como *polling*. Nesta tese, estuda-se a conveniência da utilização do escalonador baseado em *polling* para RSC. O estudo é, também, realizado através de uma plataforma de simulações que contempla fontes apropriadas desenvolvidas para sensores corporais, o escalonador *polling* e um buffer. As fontes desenvolvidas são inéditas na literatura. Os resultados mostram que o escalonador *polling* pode ser uma boa alternativa para coletar dados dos sensores sobre ou subcutâneos implantados no corpo humano.

Palavras-Chave: *Redes, Escalonador, QoS, banda efetiva, fontes.*

Abstract

Today's networks are based on IP technology and carry a variety of traffic, such as voice, data and video (multimedia traffic) and are called convergent networks. Other networks are being developed for specific applications such as Wireless Body Sensor Networks (WBSN).

The problem of providing Quality of Service (QoS) of each type of traffic in convergent networks and WBSN is essential, since the traffics demand different quality requirements. One of the main providers of QoS in these networks is the traffic scheduler. The main objective of this thesis is to analyze and propose traffic schedulers for convergent network and WBSN.

The traffic scheduler provides a fairer use of the available bandwidth. As IP based network was originally designed to carry only data traffic, it is studied in this thesis, the influence of multimedia traffic on the performance and design of the scheduler. The study is carried out by developing multiple simulation platforms that contain various types of traffics, a buffer for packet storing, an output link and the various types of scheduling.

As the FIFO scheduler was and remains the most widely used, it was taken as a reference to compare the results obtained with the DRR scheduler and with another unprecedented model of scheduling that uses in its calculation of the distribution of quotas, the concept of effective bandwidth. The DRR and the effective bandwidth schedulers can separate the traffics, so the allocation of bandwidths among the various traffics becomes fair and requirements of QoS can be met.

From the results, it can be seen that the proposed scheduling solution can control packet loss, but the delay should be better investigated. Moreover, the DRR scheduler is best suited for multimedia traffic compared to the FIFO scheduler.

In most of the proposals examined in the literature, the scheduler most used on WBSN is based on TDM technique. There are few studies in which the scheduler is based on cyclic service, known in the literature as polling. In this thesis, the convenience of using the scheduler based on polling for WBSN is studied. The study is also undertaken through a platform of simulations that include appropriate fonts developed for body sensors, the polling scheduler and a buffer. The sources developed are unprecedented in literature. The results show that the polling scheduler can be a good alternative to collect data from sensors implanted in subcutaneous or on the human body.

Keywords: Network, Scheduler, QoS, Effective Bandwidth, sources.

Sumário

AGRADECIMENTOS.....	IX
RESUMO	XI
ABSTRACT	XIII
SUMÁRIO.....	XV
LISTA DE FIGURAS	XVII
LISTA DE TABELAS.....	XIX
LISTA DE SIGLAS E ABREVIATURAS	XXI
TRABALHOS SUBMETIDOS E PUBLICADOS PELO AUTOR.....	XXV
1 INTRODUÇÃO.....	1
1.1 ORGANIZAÇÃO DO TRABALHO	4
2 REDES CONVERGENTES E ESCALONADORES DE TRÁFEGO	7
2.1 CONCEITO DE REDES CONVERGENTES.....	7
2.2 HISTÓRICO	8
2.3 BENEFÍCIOS E FATORES DE RISCO DA CONVERGÊNCIA	9
2.4 PROPOSTAS DE SOLUÇÕES PARA REDES CONVERGENTES	10
2.5 ESCALONADORES	11
2.5.1 Escalonadores para Redes Convergentes.....	12
2.5.2 Descrição de Alguns Escalonadores para Rede Convergente.....	13
2.5.2.1 Escalonador First-In, First-Out (FIFO).....	14
2.5.2.2 Escalonador com Prioridade (PQ - Priority Queueing)	19
2.5.2.3 Escalonador Weighted Round Robin (WRR).....	20
2.5.2.4 Escalonador Weighted Fair Queueing (WFQ).....	22
2.5.2.5 Escalonador Deficit Round Robin (DRR)	24
2.5.2.6 Escalonador Baseado em Banda Efetiva.....	26
2.6 CONCLUSÕES DO CAPÍTULO	29
3 MODELOS DE FONTES DE TRÁFEGO PARA REDES CONVERGENTES.....	31
3.1 MODELO DE POISSON	31
3.2 MODELO ON/OFF	32
3.2.1 Modelo On/Off de Pareto	32
3.2.2 Modelo On/Off Exponencial.....	33
3.3 MODELO AUTO-SIMILAR.....	34

3.4	JUSTIFICATIVA PARA USAR FONTES ON/OFF E PARÂMETROS PARA A SUA GERAÇÃO	35
3.5	DESCRIÇÃO DAS FONTES ON/OFF DESENVOLVIDAS	36
3.5.1	<i>Descrição da Fonte On/Off Fixa</i>	37
3.5.1.1	<i>Análise dos Resultados da Fonte On/Off Fixa</i>	38
3.5.2	<i>Descrição da Fonte On/Off Variável 1.0</i>	40
3.5.2.1	<i>Análise dos Resultados da Fonte On/Off Variável 1.0</i>	42
3.5.3	<i>Descrição da Fonte On/Off Variável 2.0</i>	43
3.5.3.1	<i>Análise dos Resultados da Fonte On/Off Variável 2.0</i>	45
3.6	CONCLUSÕES DO CAPÍTULO	46
4	ANÁLISE DE UM NÓ DA REDE CONVERGENTE COM ESCALONADORES FIFO, DRR E DE BANDA EFETIVA	49
4.1	ANÁLISE COM O ESCALONADOR FIFO	50
4.1.1	<i>Resultados dos Cenários Simulados com o Escalonador FIFO</i>	52
4.1.2	<i>Conclusões Referentes ao Escalonador FIFO</i>	60
4.2	ANÁLISE COM O ESCALONADOR DRR	61
4.2.1	<i>Resultados Comparando os Escalonadores FIFO e DRR</i>	65
4.2.2	<i>Conclusões Referentes ao Escalonador DRR Confrontado com o Escalonador FIFO</i>	82
4.3	ANÁLISE COM O ESCALONADOR BASEADO EM BANDA EFETIVA	86
4.3.1	<i>Resultados dos Cenários Simulados com o Escalonador com Banda Efetiva</i>	87
4.3.2	<i>Conclusões Referentes ao Escalonador com Banda Efetiva</i>	92
4.4	CONCLUSÕES DO CAPÍTULO	93
5	ESCALONADORES PARA REDE DE SENSORES	95
5.1	REDES DE SENSORES SEM FIOS (RSSF)	97
5.2	REDE DE SENSORES CORPORAIS	101
5.3	MODELOS DE FONTES PARA RSC	104
5.3.1	<i>Fontes Propostas para RSC</i>	106
5.3.1.1	<i>Fonte On/Off Constante</i>	107
5.3.1.2	<i>Fonte On/Off Limiar</i>	109
5.3.1.3	<i>Fonte On/Off Limiar Controlado</i>	110
5.3.1.4	<i>Fonte On/Off Fora-Faixa</i>	112
5.3.1.5	<i>Fonte On/Off Fora-Faixa Controlada</i>	114
5.3.2	<i>Análise das Fontes Propostas</i>	116
5.4	ESCALONADORES PARA REDE CORPORAL	119
5.4.1	<i>Polling</i>	122
5.5	ANÁLISE DE UMA REDE CORPORAL COM ESCALONADOR POLLING	127
5.6	CONCLUSÕES DO CAPÍTULO	137
6	CONCLUSÕES	143
6.1	TRABALHOS FUTUROS	145
	REFERÊNCIAS BIBLIOGRÁFICAS	147

Lista de Figuras

FIGURA 2.1: OPERAÇÃO DA FILA FIFO. FONTE: ADAPTADA DE [109].....	14
FIGURA 2.2: FLUXOGRAMA DO PROCESSO DE SIMULAÇÃO PARA ESCALONAMENTO FIFO. FONTE: [82]	16
FIGURA 2.3: FILAS COM PRIORIDADES. FONTE ADAPTADA DE [28].	19
FIGURA 2.4: FILAS WEIGHTED ROUND ROBIN. FONTE: [85].....	21
FIGURA 2.5: EXEMPLO DE WEIGHTED FAIR QUEUEING. FONTE: ADAPTADA DE [113]	23
FIGURA 2.6: EXEMPLO DE DÉFICIT ROUND ROBIN. FONTE: [101].	24
FIGURA 3.1: MODELO ON/OFF SIMPLES COM TAXAS DE TRANSIÇÃO T1 E T2. (ADAPTADA DE [24], (p. 05)).....	34
FIGURA 3.2: ARQUIVO DE TEXTO GERADO PELO PROGRAMA FONTE ON/OFF FIXA	38
FIGURA 3.3: ARQUIVO GERADO PELO PROGRAMA FONTE ON/OFF VARIÁVEL 1.0.....	41
FIGURA 4.1: CONFIGURAÇÃO DA REDE USADA NAS SIMULAÇÕES COM O ESCALONADOR FIFO.....	50
FIGURA 4.2: SIMULAÇÃO COM AUMENTO DE USUÁRIOS DE DADOS	53
FIGURA 4.3: SIMULAÇÃO COM AUMENTO DE USUÁRIOS DE VOZ.....	54
FIGURA 4.4: SIMULAÇÃO COM AUMENTO DE USUÁRIOS DE VÍDEO (CENÁRIO 3).....	55
FIGURA 4.5: SIMULAÇÃO COM AUMENTO DE USUÁRIOS DE VÍDEO (CENÁRIO 4).....	56
FIGURA 4.6: SIMULAÇÃO COM AUMENTO DE USUÁRIOS DE VOZ (CENÁRIO 5)	57
FIGURA 4.7: SIMULAÇÃO COM AUMENTO DE USUÁRIOS DE VOZ (CENÁRIO 6)	58
FIGURA 4.8: SIMULAÇÃO COM O AUMENTO DE USUÁRIOS DE DADOS NO ESCALONADOR DRR.....	65
FIGURA 4.9: SIMULAÇÕES COM O AUMENTO DE USUÁRIOS DE DADOS E FONTE 1.0.....	67
FIGURA 4.10: SIMULAÇÕES COM O AUMENTO DE USUÁRIOS DE DADOS E FONTE 2.0	68
FIGURA 4.11: SIMULAÇÃO COM AUMENTO DE USUÁRIOS DE VOZ E QUANTIDADE FIXA DE USUÁRIOS DE DADOS NO ESCALONADOR DRR.....	69
FIGURA 4.12: SIMULAÇÕES COM O AUMENTO DE USUÁRIOS DE VOZ E QUANTIDADE FIXA DE QUATRO USUÁRIOS DE DADOS	70
FIGURA 4.13: SIMULAÇÃO COM AUMENTO DE USUÁRIOS DE VÍDEO E QUANTIDADE FIXA DE QUATRO USUÁRIOS DE DADOS NO ESCALONADOR DRR	71
FIGURA 4.14: SIMULAÇÕES COM O AUMENTO DE USUÁRIOS DE VÍDEO E QUANTIDADE FIXA DE QUATRO USUÁRIOS DE DADOS	72
FIGURA 4.15: SIMULAÇÃO COM AUMENTO DE USUÁRIOS DE VÍDEO E QUANTIDADE FIXA DE USUÁRIOS DE VOZ E DE DADOS NO ESCALONADOR DRR	74
FIGURA 4.16: SIMULAÇÕES COM O AUMENTO DE USUÁRIOS DE VÍDEO E QUANTIDADE FIXA DE CINCO USUÁRIOS DE VOZ E DE QUATRO USUÁRIOS DE DADOS	74
FIGURA 4.17: SIMULAÇÃO COM O AUMENTO DE USUÁRIOS DE VOZ E QUANTIDADE FIXA DE USUÁRIOS DE VÍDEO E DE DADOS NO ESCALONADOR DRR	76
FIGURA 4.18: SIMULAÇÕES COM O AUMENTO DE USUÁRIOS DE VOZ E QUANTIDADE FIXA DE UM USUÁRIO DE VÍDEO E DE QUATRO USUÁRIOS DE DADOS	77
FIGURA 4.19: SIMULAÇÃO COM AUMENTO DE USUÁRIOS DE VOZ E QUANTIDADE FIXA DE DOIS USUÁRIOS DE VÍDEO E DE QUATRO USUÁRIOS DE DADOS NO ESCALONADOR DRR	78
FIGURA 4.20: SIMULAÇÕES COM O AUMENTO DE USUÁRIOS DE VOZ E QUANTIDADE FIXA DE DOIS USUÁRIOS DE VÍDEO E QUATRO USUÁRIOS DE DADOS	79
FIGURA 5.1: COMPONENTES BÁSICOS DE UM NÓ SENSOR	97
FIGURA 5.2: REDES DE SENSORES PARA O CORPO HUMANO	101
FIGURA 5.3: COMPARATIVO ENTRE A QUANTIDADE DE INTERVALOS ON	116

FIGURA 5.4: ANÁLISE DO TEMPO TOTAL DOS INTERVALOS ON.....	117
FIGURA 5.5: COMPARAÇÃO ENTRE O TOTALIZADOR DE OFF	118
FIGURA 5.6: FUNCIONAMENTO DO MECANISMO POLLING.....	125
FIGURA 5.7: PERDA DE PACOTES NO PRIMEIRO CENÁRIO.....	129
FIGURA 5.8: TEMPO DE FILA DOS PACOTES NO PRIMEIRO CENÁRIO	130
FIGURA 5.9: TEMPO DE FILA, SERVIÇO E SISTEMA NO NÓ SORVEDOURO COM O ALGORITMO FIFO E CENÁRIO 1	130
FIGURA 5.10: DESCARTE DE PACOTES DO SEGUNDO CENÁRIO.....	131
FIGURA 5.11: PERCENTUAL DE PACOTES DESCARTADOS, NO CENÁRIO 2, PELOS CRITÉRIOS RESTRITIVOS.....	132
FIGURA 5.12: TEMPO DE FILA DOS PACOTES NO SEGUNDO CENÁRIO	133
FIGURA 5.13: TEMPO DE FILA, SERVIÇO E SISTEMA NO SORVEDOURO COM O ALGORITMO FIFO NO CENÁRIO 2.	133
FIGURA 5.14: DESCARTE DE PACOTES NO TERCEIRO CENÁRIO	134
FIGURA 5.15: PERCENTUAL DE PACOTES DESCARTADOS, NO CENÁRIO 3, PELO CRITÉRIO RESTRITIVO.....	135
FIGURA 5.16: TEMPO DE FILA DOS PACOTES NO TERCEIRO CENÁRIO	136
FIGURA 5.17: TEMPO DE FILA, SERVIÇO E SISTEMA NO SORVEDOURO COM O ALGORITMO FIFO PARA O CENÁRIO 3.....	137

Lista de Tabelas

TABELA 3.1: TAXA DE PICO, TAMANHO DOS PACOTES E MÉDIA DOS INTERVALOS <i>OFF</i> DAS FONTES DE DADOS, VOZ E VÍDEO.	36
TABELA 3.2: RESULTADO DAS FONTES GERADAS PELO PROGRAMA FONTE ON/OFF FIXA	39
TABELA 3.3: RESULTADOS DAS FONTES GERADAS PELO PROGRAMA FONTE ON/OFF VARIÁVEL 1.0	42
TABELA 3.4: RESULTADOS DAS FONTES GERADAS PELO PROGRAMA FONTE ON/OFF VARIÁVEL 2.0	45
TABELA 4.1: CENÁRIOS CONSIDERADOS	51
TABELA 4.2: CENÁRIOS CONSIDERADOS PARA ANÁLISE COM O ESCALONADOR DRR	63
TABELA 4.3: TAXA DE PICO, TAMANHO DOS PACOTES E INTERVALOS DE ON E OFF DAS FONTES DE DADOS, VOZ E VÍDEO.	87
TABELA 4.4: RESULTADOS OBTIDOS COM O ESCALONADOR DRR COM QUOTA FIXA DE 512 BITS, BUFFER FINITO E COM PACOTES DE TAMANHOS MAIORES.	88
TABELA 4.5: RESULTADOS OBTIDOS COM O ESCALONADOR DRR COM QUOTA FIXA DE 512 BITS, BUFFER FINITO E COM PACOTES DE TAMANHOS MENORES.	89
TABELA 4.6: RESULTADOS OBTIDOS CONSIDERANDO O ESCALONADOR DRR COM QUOTA PROPORCIONAL À BANDA EFETIVA COM BUFFER FINITO, COM TAMANHOS MAIORES DE PACOTE.	89
TABELA 4.7: RESULTADOS OBTIDOS CONSIDERANDO O ESCALONADOR DRR COM QUOTA PROPORCIONAL A BANDA EFETIVA COM BUFFER FINITO, COM TAMANHOS MENORES DE PACOTES.	90
TABELA 4.8: RESULTADOS OBTIDOS COM O ESCALONADOR DRR COM QUOTA FIXADA EM 512 BITS E BUFFER INFINITO, COM TAMANHOS MAIORES DE PACOTE.	91
TABELA 4.9: RESULTADOS OBTIDOS COM O ESCALONADOR DRR, SENDO A QUOTA PROPORCIONAL A BANDA EFETIVA, BUFFER INFINITO, COM TAMANHOS MAIORES DE PACOTE.	91
TABELA 5.1: PARÂMETROS REFERENTES À COMUNICAÇÃO DO NÓ CENTRALIZADOR.....	99
TABELA 5.2: CONFIGURAÇÃO DO SENSOR QUANTO A ENERGIA.....	99
TABELA 5.3: CONFIGURAÇÃO DO SENSOR QUANTO À DIMENSÃO E PESO.	100
TABELA 5.4: COMPARAÇÃO ENTRE AS CONFIGURAÇÕES DOS SENSORES MICA2Dot, o MICA2 e o MICAZ.	102
TABELA 5.5: PARÂMETROS PARA A GERAÇÃO DAS FONTES	106
TABELA 5.6: RESULTADOS DAS SIMULAÇÕES COM A FONTE ON/OFF CONSTANTE.....	108
TABELA 5.7: RESULTADOS DAS SIMULAÇÕES COM A FONTE ON/OFF LIMIAR	109
TABELA 5.8: RESULTADOS DAS SIMULAÇÕES COM A FONTE ON/OFF LIMIAR CONTROLADO	111
TABELA 5.9: RESULTADOS DAS SIMULAÇÕES COM A FONTE ON/OFF FORA-FAIXA.....	113
TABELA 5.10: RESULTADOS DAS SIMULAÇÕES COM A FONTE ON/OFF FORA-FAIXA CONTROLADA	115
TABELA 5.11: CENÁRIOS SIMULADOS	128
TABELA 5.12: CARACTERÍSTICA DAS FONTES PROPOSTAS.....	139

Lista de Siglas e Abreviaturas

ϵ	Taxa de perda desejada
δ	Constante positiva chamada taxa de decaimento assintótico
$A(t)$	Número de pacotes gerados durante o intervalo de tempo $[0, t]$
ATM	<i>Asynchronous Transfer Mode</i>
bps	Bits por segundo
BRR	<i>Bit-by-bit Round Robin</i>
CAC	Controle de Admissão de Conexão
CBQ	<i>Class Based Queue</i>
CBR	<i>Constant Bit Rate</i>
CB-WFQ	<i>Class-Based Weighted Fair Queueing</i>
CQ	<i>Custom Queueing</i>
CSMA/CA	<i>Carrier Sense Multiple Access/Colision Avoidance</i>
DiffServ	<i>Differentiated Services</i>
DRR	<i>Déficit Round Robin</i>
DRR-CA	<i>Deficit Round Robin Credit Aware</i>
DTN	<i>Delay-Tolerant Network</i>
DWFQ	<i>Class-based Distributed Weighted Fair Queueing</i>
$E\{T\}$	Tempo Médio de Sistema
$E\{W\}$	Tempo Médio de Fila
$E\{X\}$	Tempo Médio de Serviço
EDD-D	<i>Earliest Due Data for Delay</i>
EDD-J	<i>Earliest Due Data for Jitter</i>
FARIMA	<i>Fractional AutoRegressive Integrated Moving Average</i>
fBM	<i>fractional Brownian Motion</i>
FCFS	<i>First-Come, First-Served</i>
FGN	<i>Fractional Gaussian Noise</i>
FIFO	<i>First In First Out</i>
Fonte 1.0	Fonte On/Off Variável 1.0
Fonte 2.0	Fonte On/Off Variável 2.0
Fonte Fixa	Fonte On/Off Fixa
FQ	<i>Fair Queueing</i>

FTP	<i>File Transfer Protocol</i>
GSM	<i>Global System for Mobile</i>
HMIPv6	<i>Hierarchical Mobile IPv6</i>
HRR	<i>Hierarchical Round Robin</i>
IETF	<i>Internet Engineering Task Force</i>
IntServ	<i>Integrated Services</i>
IP	<i>Internet Protocol</i>
k	Tamanho do buffer
LLEPS	<i>Low Latency and Efficient Packet Scheduling</i>
LRD	Dependência de Longo Alcance (<i>Long-Range Dependent</i>)
MAC	Controle de Acesso ao Meio (<i>Media Access Control</i>)
MANET	<i>Mobile Adhoc Network</i>
MMPP	Processo Poissoniano Modulado por Markov
NGN	Redes da Próxima Geração (<i>Next Generation Network</i>)
N-ISDN	<i>Narrowband Integrated Services Digital Network</i>
PCT	<i>Preemptive Cut Through</i>
PDA	<i>Personal Digital Assistants</i>
PP	Perda de Pacotes
PQ	<i>Priority Queueing</i>
PSBU	<i>Prefix Scope Binding Updates</i>
PSTN	Rede de Telefonia Pública Comutada (<i>Public Switched Telephone Network</i>)
QoS	Qualidade de Serviço
R	Taxa de geração de pacotes no período ativo
R/S	<i>Rescaled Range</i>
RCPQ	<i>Rate-Controlled Priority Queueing</i>
RED	<i>Random Early Detection</i>
RSC	Rede de Sensores Corporais
RSSF	Redes de Sensores Sem Fios
S & G	<i>Stop and GO</i>
SCFQ-CA	<i>Self-Clocked Fair Queuing Credit Aware</i>
SDH	<i>Synchronous Digital Hierarquia</i>
SLA	Acordos de Níveis de Serviço
SNA	<i>Systems Network Architecture</i>
SPQ	<i>Strict Priority Queueing</i>
SRD	Dependência de Curto Alcance
TCP	<i>Transmission Control Protocol</i>
TDM	Multiplexação por Divisão do Tempo (<i>Time Division Multiplexing</i>)
TDMA	<i>Time Division Multiple Access</i>
TI	Tecnologia da Informação
TINAc	<i>Telecommunications Information Networking Architecture Consortium</i>

TMP	Tamanho Médio dos Pacotes
T _{off}	Tempo médio do intervalo de silêncio
T _{on}	Tempo médio do período ativo
UDP	<i>User Datagram Protocol</i>
UMA	<i>Unlicensed Mobile Access</i>
VoIP	Voz sobre IP (<i>Voice over IP</i>)
WFQ	<i>Weighted Fair Queueing</i>
WINS	<i>Wireless Integrated NetWork Sensors</i>
WLAN	<i>Wireless Local Area Network</i>
WRR	<i>Weighted Round Robin</i>
WRR/SB	<i>Weighted Round Robin with Save and Borrow</i>
WRRPQ	<i>Weighted Round Robin with Priority Queueing</i>
WWW	<i>World Wide Web</i>

Trabalhos Submetidos e Publicados Pelo Autor

1. PAZETO, Tatiana A.; SILVA, Renato M.; MOTOYAMA, Shusaburo. Impacto dos Tráfegos Multimídias em um Nó de Rede Usando o Escalonador DRR In: XXXVI Conferência Latino-Americana de Informática (XXXVI CLEI) realizado em 18 a 22 de Outubro de 2010, na cidade de Assunção - Paraguai, promovido pelo Centro Latino Americano de Estudos em Informática.
2. PAZETO, Tatiana A.; SILVA, Renato M.; MOTOYAMA, Shusaburo. Impact of the Multimedia Traffic Sources in a Network Node Using FIFO scheduler. In: Networked Digital Technologies. Praga, Republica Tcheca. 2010. Communications in Computer and Information Science, 2010, Volume 87, Part 6, pp. 545-555, DOI: 10.1007/978-3-642-14292-5_56.
3. PAZETO, Tatiana A.; SILVA, Renato M.; MOTOYAMA, Shusaburo. Impacto das Fontes de Tráfego Multimídias em um Nó de Rede Usando o Escalonador FIFO. In: III Congresso Tecnológico TI e Telecom (InfoBrasil 2010), realizado em Fortaleza-CE, de 26 a 28 de maio de 2010.
4. SILVA, Renato Moraes; PAZETO, Tatiana Annoni. Proposals of On/Off Sources and their effects on the DRR and FIFO schedulers, indicating the most influential parameter. In: International Conference on Computer Communications and Network (CCN-10), realizado de 12 a 14 de julho de 2010 em Orlando, Florida, USA. Esta conferência faz parte do MULTICONF-10. ISBN: 978-1-60651-018-6.
5. PAZETO, Tatiana A.; MOTOYAMA, Shusaburo. QoS Guaranteed Traffic Scheduling in Convergent Networks Using Effective Bandwidth. In: International Workshop on Telecommunications. (IWT'07) Santa Rita do Sapucaí, MG. 2007.
6. PAZETO, Tatiana A.; MOTOYAMA, Shusaburo. Escalonamento de Tráfego Usando a Banda Efetiva de Kesidis para Garantir QoS em Redes Convergentes. In: 5th International Information and Telecommunication Technologies Symposium (I2TS 2006), que ocorreu em Cuiabá no período de 06 a 08 de dezembro de 2006.
7. REFATTI, Luis Fernando; PAZETO, Tatiana Annoni. Análise e Impacto das Fontes para Redes de Sensores para o Corpo Humano. Revista Brasileira de Inovação Tecnológica em Saúde (R-BITS). 2011. Disponível em: <http://www.incubadora.ufrn.br/incubadora/index.php>

/reb/index. A R-BITS está registrada no LATINDEX – sistema Regional de Información en Línea para Revistas Científicas da América Latina, el Caribe, España y Portugal – <http://www.latindex.unam.mx>.

8. PAZETO, Tatiana A.; REFATTI, Luís F.; MOTOYAMA, Shusaburo. Polling-based Medium Access Control Scheme for Wireless Body Sensor Network. In: 11th International Conference on Wireless Networks (ICWN'12), realizado de 16 a 19 de julho de 2012 em Las Vegas, Nevada, USA. Esta conferência faz parte do The 2012 World Congress in Computer Science, Computer Engineering, and Applied Computing List of Joint Conferences (WORLDCOMP'12).

1 INTRODUÇÃO

Até recentemente, a maioria dos tráfegos transportados em redes de computadores era restrito ao tráfego de dados. Contudo, com a disseminação da Internet, novos serviços foram sendo oferecidos, proporcionando melhorias para a comunidade, principalmente no que se refere a diversões e facilidade na comunicação entre as pessoas. Desta forma, todos os tipos de tráfego, como voz, vídeo, imagens e dados trafegam em uma rede. Assim, o transporte desses tráfegos, denominado neste trabalho de multimídia, pode causar um grande impacto sobre a concepção e dimensionamento de um nó de rede.

Para a integração de todos os tipos de tráfego, a rede IP está sendo usada como plataforma de convergência. Entretanto, a rede IP atual, por exemplo, a Internet, não está preparada para satisfazer a Qualidade de Serviço (QoS) de diferentes tráfegos tais como voz, vídeo e dados. Com o aumento cada vez maior das aplicações, a necessidade da plataforma IP de satisfazer a QoS de cada aplicação tornou-se imperativo.

Os quatro princípios para prover QoS são [65] a classificação de pacotes, isolamento de tráfegos (fluxos) e regulação, alto índice de utilização de recursos e aceitação de chamadas. A *Internet Engineering Task Force* (IETF) propôs o DiffServ [14] com o objetivo de satisfazer a QoS. Na solução DiffServ, os pacotes são classificados em classes de fluxo. Cada classe de fluxo pode ser tratada de uma mesma maneira na rede inteira. Neste tipo de rede, o usuário faz um acordo com o provedor de serviços que estabelece alguns parâmetros do contrato, tais como a taxa média de tráfego, a taxa de pico do tráfego e a perda de pacotes. Baseado em tais acordos, o provedor de

serviços deve fornecer para todos os usuários, individualmente, a QoS contratada. Para o provedor, além de garantir os acordos, é necessário o uso eficiente dos recursos da rede. Porém, de acordo com o segundo princípio de QoS, é idealizado o isolamento entre os tráfegos, de tal forma que haja justiça entre as classes de fluxo no uso dos recursos.

Assim, uma forma para realizar o tratamento dos pacotes de modo a garantir a equidade e diferenciação entre os fluxos é usar algoritmos de escalonamento, sendo que o DiffServ não propõe nenhum tipo de escalonador para tratamento desses tráfegos. Há várias soluções propostas na literatura, que tentam resolver, principalmente, o problema de equidade dos tráfegos tais como o *Weighted Round Robin* (WRR), *Weighted Fair Queueing* (WFQ) [33] e *Deficit Round Robin* (DRR) [118], [111].

Um dos objetivos deste trabalho é comparar o desempenho dos escalonadores FIFO e DRR para as redes convergentes multimídia baseados na arquitetura IP. O escalonador FIFO é o primeiro a ser analisado por ser o mais difundido e estar presente na maioria dos equipamentos de interconexão. Contudo, o FIFO tem como único critério para atender o tráfego, o tempo de chegada dos pacotes, não possibilitando garantir QoS e distribuir os recursos de forma justa. Além disso, ele foi projetado para tratar apenas um tipo de tráfego, e pode não ser apropriado para atender aplicação multimídia. Já o DRR foi escolhido e analisado em função da sua simplicidade na implementação e por permitir priorizar certos tráfegos, atribuindo quotas diferenciadas, possibilitando atender a QoS e garantir a equidade na distribuição dos recursos.

Além disso, um novo modelo de escalonamento é proposto, que além da equidade dos tráfegos, tem o objetivo de satisfazer a QoS de cada tráfego nas redes convergentes. Nessa proposta é utilizado o conceito de banda efetiva ou banda equivalente. A banda efetiva é uma banda fixa estimada com base nos parâmetros do Acordo de Nível de Serviço (SLA – *Service Level Agreement*). Portanto, é uma banda que leva em consideração a QoS do tráfego. Vários algoritmos para cálculo de banda efetiva são propostos na literatura. Para o tráfego correlacionado baseado no modelo On/Off, mas não dependente de longo intervalo, LDR (*Long Range Dependence*), a largura de banda efetiva pode ser estimada usando os cálculos apresentados em [62] e [50]. No caso do tráfego auto-

similar com LDR a largura de banda efetiva pode ser estimada usando o cálculo apresentado em [45].

No esquema de escalonamento de tráfego proposto, o serviço para cada tráfego, em um ciclo, é proporcional à largura de banda efetiva. A utilização desse esquema permite a implementação de um escalonador muito simples com a garantia de QoS para cada tráfego. O funcionamento da proposta é exemplificado através da utilização da banda efetiva apresentada em [62].

O estudo de desempenho dos escalonadores acima mencionados foi realizado, neste trabalho, através de simulações. Assim, inicialmente, duas plataformas de simulação foram desenvolvidas em C++. A primeira contém o escalonador FIFO, um buffer e diferentes tipos de fontes de tráfego. A segunda contempla o escalonador DRR, um buffer e as mesmas fontes de tráfego da plataforma anterior.

Vários cenários foram analisados, sendo que o primeiro se restringe somente ao tráfego de dados, comparando com os outros cenários, que misturam tráfegos de dados, voz e vídeo. O intuito foi verificar como os tráfegos multimídias podem impactar a rede atual, bem como verificar o comportamento de um nó da rede em relação ao tempo de atraso de sistema que é um dos parâmetros importantes na definição da qualidade de serviço para os usuários.

Já a plataforma com o escalonador de banda efetiva proposto foi desenvolvida em MatLab e visa analisar se este novo algoritmo de escalonamento consegue reduzir a perda de pacotes e o atraso no sistema.

A tese também tem como objetivo analisar o desempenho do escalonador de serviço cíclico (polling) para redes de sensores corporais sem fio – RSC (ou WBAN – *Wireless Body Area Network*). O escalonador de serviço cíclico é utilizado no esquema de controle de acesso ao meio (MAC), para coletar os dados em tempo quase real dos sensores colocados no corpo humano. Os principais parâmetros utilizados no estudo são a perda de pacotes e o tempo de espera no buffer de um nó

sensor. Uma vez que o nó sensor para Redes de Sensores Corporais (RSC) necessita poupar energia, é também estudado o tamanho mínimo do buffer para manter os pacotes antes de sua transmissão.

Para atingir os objetivos acima, uma plataforma de simulação foi desenvolvida em C++, contemplando o mecanismo polling, o escalonador FIFO e as fontes dos sensores. Devido à falta de modelos de nó sensor para aplicações de RSC na literatura, cinco fontes de sensores foram desenvolvidas, todas baseadas no modelo On/Off.

1.1 ORGANIZAÇÃO DO TRABALHO

O trabalho está organizado em seis capítulos, sendo que o primeiro apresenta a introdução ao assunto do referido trabalho e a estrutura da tese.

O segundo capítulo versa sobre redes convergentes, relatando seu conceito, um breve histórico e benefícios, bem como o uso da rede IP como plataforma de convergência. Também são abordados algoritmos de escalonamento para as redes convergentes, iniciando com a definição de escalonador. Em seguida, alguns algoritmos encontrados na bibliografia são detalhados, explanando seu funcionamento, vantagens e desvantagens.

O capítulo seguinte consiste em apresentar os motivos da utilização de fontes de tráfego, bem como alguns modelos já consolidados na literatura. O capítulo culmina com a justificativa para o uso de modelos de fontes On/Off Exponencial para redes convergentes, apresentando três propostas e analisando o comportamento das mesmas.

O próximo capítulo analisa um nó da rede convergente com os escalonadores FIFO, DRR e a proposta inédita de escalonamento que usa o cálculo da banda efetiva proposta [62] para a distribuição dos recursos entre os tráfegos. O intuito é verificar qual dos três algoritmos de escalonamento é mais indicado para o tráfego multimídia, que está muito presente nas redes atuais.

O capítulo cinco versa sobre Redes de Sensores Corporais (RSC), apresentando sua definição e funcionamento, bem como alguns parâmetros de sensores para estas redes. Modelos de fontes e

escalonadores para as RSC também são assuntos contemplados no capítulo, apresentando quatro propostas de fontes com suas respectivas análises. Além disso, uma plataforma de simulação com o escalonamento polling juntamente com as fontes e os buffers foi desenvolvida para analisar o desempenho do escalonador em redes corporais.

No capítulo seis são apresentadas as conclusões obtidas a partir dos estudos realizados, bem como sugestões para trabalhos futuros. Finalizando, encontram-se as referências bibliográficas utilizadas para realizar esta tese.

2 REDES CONVERGENTES E ESCALONADORES DE TRÁFEGO

Atualmente, os vários tipos de tráfegos, tais como voz, vídeo e dados, estão sendo transportados em uma única rede, utilizando a tecnologia IP, e está sendo denominada de rede convergente. Essa rede convergente que transporta uma diversidade de tráfegos necessita resolver vários problemas que antes foram solucionados utilizando tecnologias específicas para cada tipo de tráfego. Um dos principais problemas que a rede convergente precisa tratar se refere ao provisionamento de qualidade de serviço (QoS) dos tráfegos transportados. Uma das técnicas para provisionar a QoS em redes convergentes é a utilização de escalonadores de tráfego nas saídas dos nós da rede. Um escalonamento adequado dos tráfegos permite uma distribuição racional dos recursos existentes, além de garantir a QoS de cada tipo de tráfego.

Neste capítulo, serão descritos alguns fatos históricos e técnicos que levaram ao conceito de rede convergente e a descrição dos principais escalonadores existentes na literatura e a proposta de um novo escalonador baseado em banda efetiva.

2.1 CONCEITO DE REDES CONVERGENTES

A convergência de redes pode ser definida como um processo onde as telecomunicações, a tecnologia de informação e os meios de comunicação começam a trabalhar juntos. Uma consequência da junção é a redução de custos nos níveis de infra-estrutura, dispositivos de usuários finais e serviços [34].

As redes convergentes podem ser utilizadas para serviços fixos e móveis. Além da possibilidade de acessar serviços de tecnologias diferentes num mesmo terminal, as redes convergentes permitem o uso de aplicações que contemplam voz, dados e vídeo.

A convergência é um padrão geral no processo evolucionário, no qual a tendência é oferecer diversos serviços em uma única rede/meio ou o mesmo serviço em mais de uma rede/meio [52].

2.2 HISTÓRICO

Inicialmente o desenvolvimento das telecomunicações e da computação eram independentes, porém se tornaram relacionadas e mutuamente dependentes [52].

A metade da década de 70 é o ponto de partida. Os circuitos integrados e microprocessadores permitiram a mudança do analógico para a transmissão digital.

Entre 1974-1983 houve diversos desenvolvimentos importantes. Como os princípios da comutação de pacotes já estavam estabelecidos, foi desenvolvido o padrão X.25 e, no final deste período, o TCP/IP foi adotado como base da ARPANET, a antecessora da Internet.

O conceito de uma única rede provendo tanto serviços de voz como de dados, gerou a primeira rede com múltiplos serviços utilizando os padrões do *Narrowband Integrated Services Digital Network* (N-ISDN) [52]. Nesse período também foi desenvolvido o primeiro cabo de fibra óptica, bem como a primeira geração de redes móveis (analógicas) começou a operar.

Na próxima década, 1984-1993, houve significativos avanços tecnológicos e regulamentares. Existia cerca de mil hosts na Internet no início do período. A telefonia digital passou a fazer parte de redes públicas e privadas. Os padrões para comutação de pacotes se expandiram, passando a incluir o *Asynchronous Transfer Mode* (ATM) e Frame Relay. Foi lançada a *World Wide Web* (WWW) e o número de hosts cresceu para um milhão no final do período. Também foi padronizada e lançada a segunda geração de redes móveis *Global System for Mobile* (GSM).

Entre os anos de 1994 e 2003, ocorreu o lançamento comercial de provedores de serviços de Internet além do desenvolvimento de padrões para a telefonia utilizando redes IP. O conceito de uma rede multisserviço nova foi formulado com a emissão da primeira licença de rede móvel de terceira geração (3G). Porém o excesso de taxas de licença e a recessão econômica limitaram a implantação. Também houve um aumento na capacidade das transmissões por fibra óptica devido ao uso de múltiplos comprimentos de onda em uma única fibra, expandindo a Internet para mais de 100 milhões de hosts.

Nos últimos tempos, a tecnologia *Wireless Local Area Network* (WLAN), baseada na família 802.11, se tornou um recurso na solução de necessidades de comunicação em múltiplas aplicações [35]. Ganharam maiores destaques as aplicações relacionadas às comunicações que não somente transportam dados, ou algum tipo de navegação e acesso à Internet, mas principalmente comunicação de voz baseada em IP e vídeo [35].

2.3 BENEFÍCIOS E FATORES DE RISCO DA CONVERGÊNCIA

Existem benefícios e fatores de risco com a convergência. Entre os benefícios está a redução nos custos de implementação e manutenção de redes distintas para cada serviço, como também a redução de custos da conta telefônica, utilizando serviços de voz na rede de dados, como no caso do VoIP [37].

Outras vantagens são o aumento na agilidade, a flexibilidade para se adaptar ao crescente mundo dos negócios e a redução na complexidade operacional e da rede [35] e [125].

Em relação a risco, em [37] é citada a falta de profissionais qualificados para a implementação e manutenção desse tipo de rede, a perda da qualidade da voz e menor segurança em relação a redes de voz.

2.4 PROPOSTAS DE SOLUÇÕES PARA REDES CONVERGENTES

Uma das primeiras propostas de redes convergentes foi denominada de Rede Digital de Serviços Integrados de Faixa Estreita (RDSI-FE). Foi uma tecnologia baseada em comutação de circuito e foi uma tentativa de adaptar a tecnologia muito utilizada em telefonia para integrar alguns tipos de serviços, tais como voz e dados. Para integrar uma maior variedade de tráfegos foi proposta a Rede Digital de Serviços Integrados de Faixa Larga (RDSI-FL) em que qualquer tipo de tráfego poderia ser transportado. A tecnologia utilizada foi o *Asynchronous Transfer Mode* (ATM). O ATM utiliza a técnica de comutação de pacote baseada em circuito virtual, e um pacote tem um comprimento de somente 53 bytes, denominado de célula.

Um esforço muito grande de padronização do ATM foi feito, tanto pelas indústrias como por órgãos de normatização. Entretanto, devido à demora na padronização, na utilização de muito *overhead* de comunicação devido ao tamanho pequeno de pacote, na necessidade de desenvolvimento de novos equipamentos específicos para o ATM, essa tecnologia foi suplantada em favor da tecnologia IP.

As razões da utilização da tecnologia IP em redes convergentes são muitas. Tem baixo custo e habilidade para combinar a mobilidade e a rede IP fixa [11] e [79]. Permite a transferência de um grande número de serviços como voz, dados e vídeo através da mesma rede e reduz a complexidade operacional da rede, resultando em um serviço melhor e mais confiável [35].

A rede IP é baseada no princípio fim a fim, onde o núcleo da rede deve ser o mais simples possível, deixando a complexidade para as bordas. A rede IP tem um design modular, onde cada camada implementa um serviço específico. Isso permite que o IP seja executado sobre qualquer tecnologia MAC e PHY, embora isso não seja aplicado às redes sem fio [79].

Em redes IP fixas as falhas são consideradas incomuns, uma vez que os equipamentos para redes fixas são extremamente confiáveis. Por exemplo, a probabilidade de um bit ser transmitido com erro em um cabo de fibra óptica é de 10^{-10} . Entretanto, em comunicação sem fio, a taxa de erro

pode chegar a 10^{-3} , uma vez que os nós sem fio podem falhar por falta de energia, pouca qualidade na transmissão, ou até porque o nó saiu do alcance da rede [79].

Para resolver os problemas em relação às redes sem fio, em [79] são apresentadas algumas soluções, como o MobileIPv6, em que cada estação móvel tem um endereço universal, que é um alias para seu IP instantâneo e o padrão IEEE 802.21 que tenta definir interfaces abertas no contexto de handover entre a família 802 de tecnologias sem fio e o 3GPP e, também, a interoperabilidade entre os tipos de rede heterogêneos [32].

Outra proposta mais radical é apresentada em [17], denominada Projeto NewArch em que propõe uma nova arquitetura para a Internet, assumindo o *redesign* desde o início da rede. O projeto Plutarch [31], [79] faz uma abordagem contrária à NewArch, propondo uma adaptação da arquitetura, permitindo a comunicação entre diferentes tecnologias de rede. Interligando, dessa maneira, as redes heterógenas e possibilitando que cada domínio utilize os protocolos que melhor se adequem às necessidades. Um dos problemas dessa proposta é que ela não permite a mobilidade dos terminais.

2.5 ESCALONADORES

A convergência de diferentes tipos de tráfegos com diversos tipos de prioridades são uma realidade nas redes de computadores atuais. Logo, para garantir a exigência das aplicações é necessário preocupar-se com os critérios de QoS para que a banda seja distribuída de forma justa.

Sendo limitados, os recursos disponíveis devem ser utilizados da forma mais eficiente possível. Uma rede congestionada pode causar a diminuição da sua vazão, o aumento do atraso e a excessiva perda de pacotes, afetando diretamente a QoS do tráfego. Deste modo, é necessário utilizar alguns métodos de controle de admissão e distribuição de recursos [106].

Uma alternativa para gerenciar a utilização dos recursos é o uso de sistema de filas. As filas são formadas pelos pacotes e pelos canais de serviço que representam o meio de transmissão. Os

pacotes chegam à fila e fazem a requisição de um serviço, como a transmissão. Caso não possa ser imediatamente atendido, o pacote fica na espera até o momento em que haja disponibilidade de atendimento.

A forma de gerenciamento de uma ou mais filas é feita através de escalonamento. O escalonamento controla o acesso do pacote à banda, decidindo quantos pacotes aceitar e a que taxa de transmissão na entrada de uma interface, bem como quais os pacotes a transmitir e em que ordem na saída.

O escalonamento de pacotes é um processo realizado por um algoritmo implementado em dispositivos de rede. Alguns escalonadores têm como objetivo criar uma fila de prioridades para o atendimento de pacotes de tal modo que evite congestionamentos e consequentemente atrasos, além da perda de pacotes. Caso haja a perda de pacotes é necessário o reenvio das informações o que ocasiona o não atendimento da QoS requerida, bem como deixa de maximizar, a longo prazo, a taxa de rendimento médio da rede, com evasão de recursos [25].

Um algoritmo de escalonamento é um programa que deve fornecer um bom desempenho em termos de alocação de largura de banda, e que satisfaça outras propriedades importantes, como ter um baixo custo computacional e complexidade de implementação reduzida [84]. Para realizar esta tarefa, vários algoritmos de escalonamento foram propostos na literatura, sendo que alguns dos mais disseminados para trabalhar em redes convergentes serão abordados, enfatizando seu funcionamento.

2.5.1 Escalonadores para Redes Convergentes

Com o crescimento de aplicações em tempo real e aplicações multimídia, como voz sobre IP e vídeo, algoritmos de escalonamento em switches e roteadores que garantam limites de desempenho para satisfazer a QoS em diferentes serviços são de extrema importância [131].

Observa-se que a distribuição da largura de banda deve promover a equidade entre as sessões, sendo que os escalonadores de pacotes têm sido considerados os principais componentes do processo [59]. Além disso, ao longo da evolução da Internet, a QoS e a justiça no acesso, sempre foram salientadas [7].

O escalonador *Weighted Fair Queueing* (WFQ) é apresentado em [7] como um dos escalonadores mais utilizados em redes convergentes. Porém, o WFQ não foi utilizado em seu trabalho, devido à complexidade quanto à implementação. Assim, adotaram uma implementação mais simples e rentável, que apresenta enfileiramento justo, sendo usado o algoritmo *Déficit Round Robin* (DRR).

Outros algoritmos utilizados em redes convergentes como: *Earliest Due Data for Delay* (EDD-D), *Earliest Due Data for Jitter* (EDD-J), *Preemptive Cut Through* (PCT), *Stop and Go* (S & G) e *Hierarchical Round Robin* (HRR) são citados em [9].

2.5.2 Descrição de Alguns Escalonadores para Rede Convergente

A seguir são descritos alguns escalonadores que podem ser utilizados em redes convergentes e a proposta de um escalonador que pode ser adequado para tratamento de tráfegos multimídias.

Os principais algoritmos que podem ser convenientes para redes convergentes são aqueles que tentam solucionar o problema de equidade dos tráfegos tais como o *Weighted Round Robin* (WRR), *Weighted Fair Queueing* (WFQ) e *Deficit Round Robin* (DRR). Além desses algoritmos FIFO e PQ serão, também, descritos.

Junto com as descrições dos algoritmos, serão dadas justificativas para a escolha do escalonador que será utilizado nesta tese.

2.5.2.1 Escalonador First-In, First-Out (FIFO)

Também chamado de *First-Come, First-Served* (FCFS), o FIFO é um escalador do tipo sem classe. É um dos algoritmos mais simples e o mais utilizado, onde o primeiro a entrar é o primeiro a sair, ocorrendo descarte de pacotes quando a fila está cheia [90]. Além disso, é o escalonador mais popular [46].

A simplicidade computacional do algoritmo FIFO faz com que ele seja um dos mais implementados nos equipamentos de interconexão [135].

A partir de seu funcionamento constata-se que não há nenhum tipo de priorização ou diferenciação do tráfego, sendo que os pacotes que chegam são alocados em uma única fila de onde são enviados para o destino na ordem em que chegaram.

Como em escalonadores do tipo FIFO não existe qualquer classificação de tráfego, pode haver consumo total da banda disponível ou provocar atrasos ou perda de pacotes importantes [28]. Além disso, o tráfego em rajada pode causar longos atrasos em aplicações sensíveis ao tempo [29]. Por este motivo, filas FIFO não servem para aplicações que necessitam de QoS [28], [85].

A Figura 2.1 demonstra o funcionamento do enfileiramento FIFO.

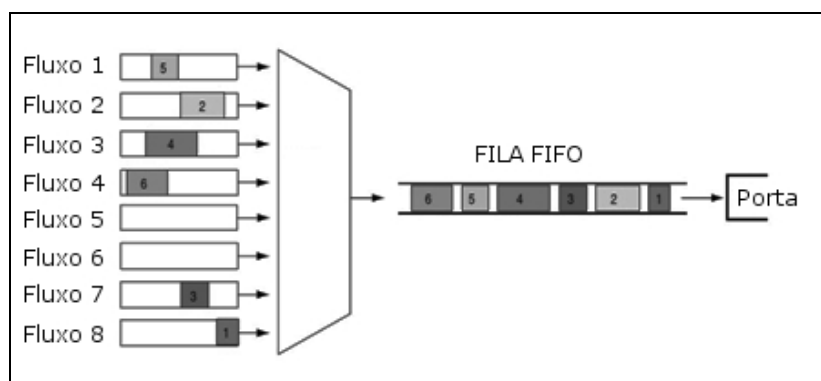


Figura 2.1: Operação da fila FIFO. Fonte: Adaptada de [109]

A Figura 2.1 mostra o funcionamento genérico de um escalonador FIFO. Pela Figura 2.1, observam-se oito filas de onde chegam os pacotes para serem processados. Os pacotes estão enumerados de acordo com a sua ordem de chegada. No exemplo da Figura 2.1, o pacote que está

na fila oito (8) foi o primeiro a chegar, sendo o primeiro a ser processado e colocado na fila FIFO de saída. Após o seu processamento, o pacote da fila dois (2) será inserido na fila, e assim por diante, até chegar ao pacote da fila quatro (4), o número seis (6), que foi o último a chegar.

O FIFO é pouco eficiente, uma vez que não cumpre com as expectativas que pairam sobre escalonadores nos dias de hoje. Isso ocorre, pois não basta somente organizar a entrada e saída. Outras funcionalidades são esperadas. Apesar disso, o mesmo apresenta algumas vantagens, a citar [109]:

- para roteadores baseados em software, este é um algoritmo que não sobrecarrega as máquinas se comparado a outros escalonadores mais sofisticados;
- como os pacotes não são reordenados, seu comportamento é previsível;
- o atraso máximo é determinado pelo tamanho da fila de espera. Desta forma, quanto menos pacotes na fila, menor será o tempo de fila em cada nó da rede.

Algumas limitações do escalonador FIFO são [109]:

- como há uma única fila, não permite organizar os pacotes em diferentes categorias de forma a oferecer um atendimento independente das outras classes;
- tendo uma só fila, o atraso influencia todos os fluxos;
- em caso de congestionamento na rede, as aplicações que usam *User Datagram Protocol* (UDP) na camada de transporte são favorecidas em relação as que usam *Transmission Control Protocol* (TCP), tendo em vista que estas são notificadas das perdas de pacotes, reduzindo a sua taxa de transmissão, aumentando o atraso e o jitter;
- um fluxo pode consumir toda a memória de uma fila de espera FIFO, provocando a negação de serviço aos demais fluxos. Isso gera aumento do atraso, jitter, e perdas para todos os outros fluxos.

Em [28] é mencionado que o escalonador FIFO é o escalonador mais rápido, sendo eficaz em grandes links, nos quais há pouco atraso e congestionamento. Por estas razões, este será um dos algoritmos de escalonamento implementado, sendo o fluxograma para o desenvolvimento do simulador apresentado na Figura 2.2.

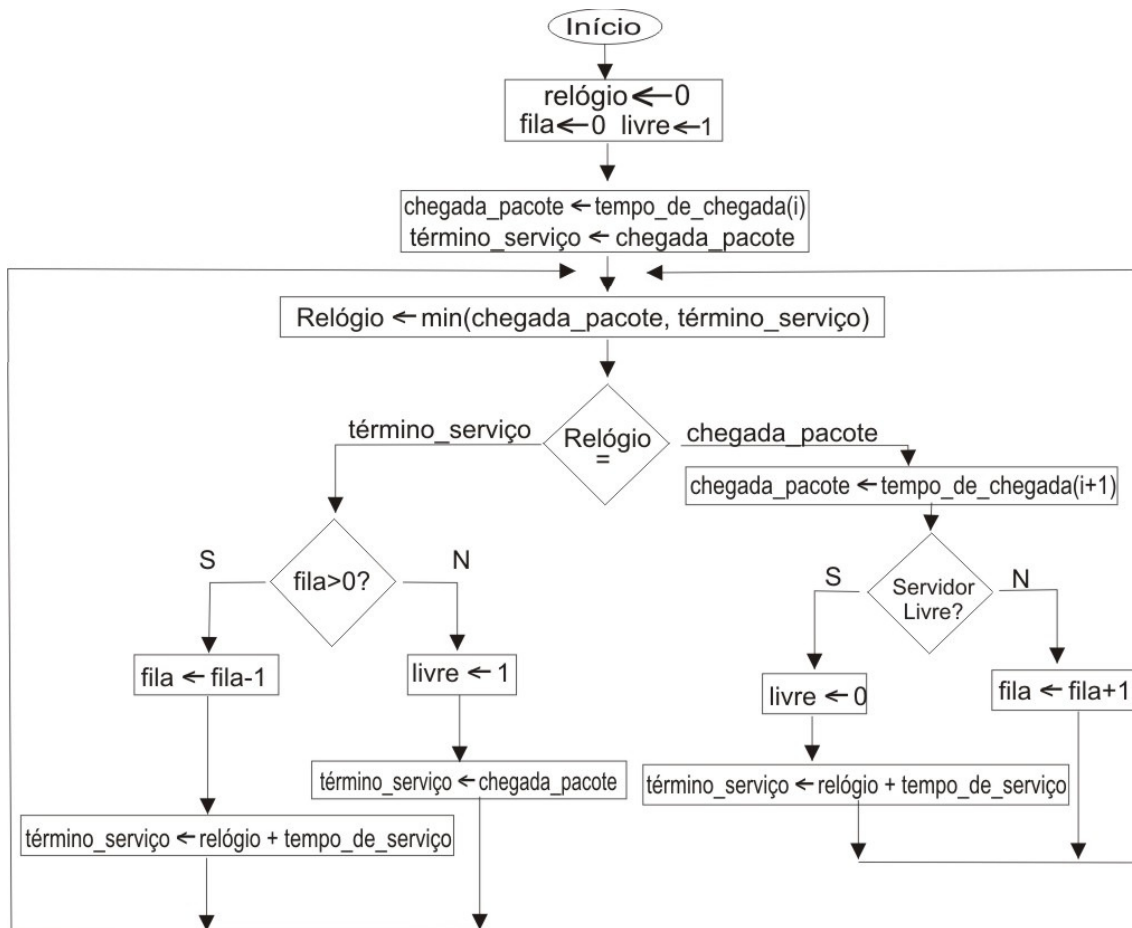


Figura 2.2: Fluxograma do processo de simulação para escalonamento FIFO. Fonte: [82]

Como pode ser observado na Figura 2.2, o simulador utiliza a técnica de eventos discretos. Os dois principais eventos são a chegada de pacotes e o término de serviço, representados, respectivamente, pelas variáveis *chegada_pacote* e *término_serviço*. Como o próprio nome já revela, o evento *chegada_pacote* serve para armazenar o tempo de chegada do pacote a ser processado. Já a

variável *término_serviço* guarda o pacote que deve ser tratado na saída do escalonador. Deste modo, o simulador sempre estará manipulando uma destas variáveis.

Para decidir se irá ocorrer primeiro o evento *término_serviço* ou *chegada_pacote*, é observado o valor das duas variáveis. Se o *término_serviço* for menor, este processo será executado. Caso contrário será realizado o processo *chegada_pacote*.

Três principais variáveis são utilizadas: fila, livre e relógio. A variável relógio armazena o tempo de chegada do pacote ou o término do serviço. A variável fila armazena o número de pacotes na fila, e a variável livre controla o estado do servidor, indicando se está pronto para atender um pacote, representado pelo número um (1), ou está ocupado, indicado pelo valor zero (0).

As variáveis fila, livre e relógio são inicializadas com valor zero.

No início do programa, só existe a possibilidade de ocorrer o evento chegada de pacote, já que não existe ainda nenhum pacote no sistema. Desta forma, *chegada_pacote* irá receber o tempo de chegada do primeiro pacote, representado pela variável *tempo_chegada(i)*. A variável relógio também é atualizada, recebendo o valor da variável *chegada_pacote*. Após isso, a variável *chegada_pacote* já recebe o valor do tempo de chegada do próximo pacote, pois como só pode ocorrer um evento por vez, então o programa já precisa conhecer o tempo de chegada do próximo pacote, para decidir qual evento irá processar depois do tratamento do pacote atual.

Como mencionado, no início da simulação a variável livre receberá o valor um (1), indicando que o servidor está disponível para atender ao pacote. Desta forma, ao receber o pacote esta variável receberá o valor zero (0), pois o servidor passará para o estado de ocupado. Como não existe nenhum pacote na fila, então o primeiro pacote já será tratado na chegada e desta forma seu tempo de espera será zero. Logo, a variável *término_serviço* receberá a soma do valor atual do relógio acrescido ao tempo de serviço do pacote. Este último valor corresponde ao tamanho do pacote dividido pela capacidade do canal.

Depois disso, será comparado o valor do término de serviço do primeiro pacote e o tempo de chegada do segundo pacote, sendo que o evento que possuir menor tempo será executado na sequência.

Supondo que a variável *término_serviço* possua menor valor, então este evento será executado pelo simulador. Desta forma, o relógio será atualizado com o valor do término de serviço. Depois disso, como não existe nenhum pacote na fila de espera, a variável livre receberá o valor um (1), para indicar que o servidor está livre e o término de serviço será atualizado com o valor do tempo de chegada do segundo pacote.

O próximo passo será verificar qual evento deve ocorrer, o término de serviço ou a chegada de pacote. Como o primeiro pacote já foi atendido e processado na saída e não existe nenhum pacote no sistema, então só existe a possibilidade de ocorrer o evento *chegada_pacote*. Pelo mesmo motivo, o segundo pacote já será processado e seu tempo de espera possuirá valor zero. Logo, o servidor passará ao estado de ocupado e a variável *término_serviço* receberá a soma entre o relógio e o tempo de serviço do pacote atual.

Novamente será comparado o valor do término de serviço do pacote atual ao valor do tempo de chegada do próximo pacote. Supondo que este último possua menor valor, então o relógio será atualizado com o valor da variável *chegada_pacote*. Diferente das situações anteriores, desta vez o servidor se encontra ocupado processando o segundo pacote. Assim, o sistema irá avaliar se existe espaço na fila ou não. Caso exista, o número de pacotes na fila será incrementado. Se não existir, o pacote será descartado.

Após isso, deve-se determinar qual evento deve ser realizado: o término de serviço do segundo pacote, que ainda não teve sua saída processada, ou o tempo de chegada do quarto pacote. Se término de serviço for menor, então este processo será realizado. Logo, como não foi feito o término de serviço do segundo pacote, isso será realizado e o servidor passará ao estado de livre.

Diante disso, será comparado o valor do tempo de chegada do quarto pacote com o valor do término de serviço do terceiro. Supondo que o valor da segunda variável seja menor, então será

realizado o evento *término_serviço*. Logo, o sistema irá verificar se existem pacotes na fila ou não. Como o terceiro pacote se encontra na fila, então ele será processado e a variável fila será decrementada. Como não existe mais nenhum pacote na fila de espera, então o evento *término_serviço* não pode ocorrer. Então, a variável *término_serviço* receberá o valor do tempo de chegada do quarto pacote. Desta forma a chegada do quarto pacote será processada e os procedimentos descritos serão executados até que o último pacote seja tratado pelo simulador.

2.5.2.2 Escalonador com Prioridade (PQ - Priority Queueing)

Denominado fila com prioridade, os tráfegos são divididos, por exemplo, em quatro níveis de prioridade: alta, média, normal e baixa. Os pacotes não classificados são marcados, por default, como normal [28], [113]. Os pacotes são primeiramente classificados e encaminhados à fila correspondente. As filas com maior prioridade têm preferência durante a transmissão, sendo atendidas por completo [28]. Para os tráfegos de mesma prioridade o atendimento dos pacotes pode ser feito pelo método FIFO [65]. Pode ocorrer o não envio de mensagens com prioridade baixa, devido ao consumo de toda a banda pelos pacotes de prioridade superior [28].

Este tipo de escalonador permite classificar o tráfego de várias formas, entre elas destacam-se: por interface de entrada; por protocolo; por lista de acesso utilizando o endereço IP do cliente, endereço *Media Access Control* (MAC); listas de usuários; tamanho de pacotes [28], [85].

Na Figura 2.3 é apresentado o funcionamento do enfileiramento PQ.

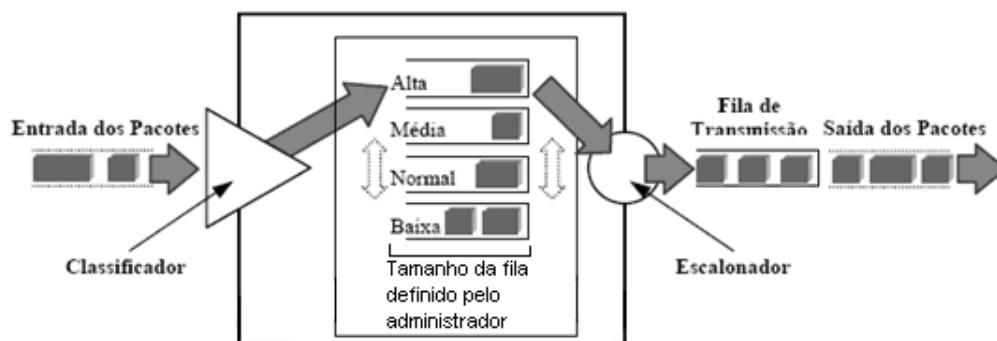


Figura 2.3: Filas com Prioridades. Fonte adaptada de [28].

O escalonador PQ é o que permite implementar mais facilmente a diferenciação dos tipos de tráfego. No entanto, caso não haja nenhum mecanismo de Controle de Admissão de Conexão (CAC) para fluxos de maior prioridade, pode ocorrer negação de serviço aos tráfegos de prioridade inferior - fenômeno designado por *starvation* [128]. Contudo, isto pode ser prevenido utilizando outro esquema de escalonamento simultaneamente, de forma que possa ser usado em redes convergentes.

O escalonamento PQ pode ser configurado de duas maneiras [64], [87]:

- *Strict Priority Queueing* (SPQ): esta configuração garante que os pacotes de uma fila de alta prioridade sempre sejam atendidos antes das filas de prioridade mais baixa. Contudo, pode haver uma saturação da banda e perda de pacotes menos prioritários se ocorrer excesso de tráfego de alta prioridade;
- *Rate-Controlled Priority Queueing* (RCPQ): garante que o tráfego de alta prioridade seja atendido antes de pacotes de menor prioridade somente se a quantidade de tráfego de alta prioridade permanecer abaixo de um limite que pode ser configurado pelo usuário. A dificuldade em estabelecer esse limite está no desconhecimento do tamanho dos pacotes, do volume e do comportamento das classes.

2.5.2.3 Escalonador Weighted Round Robin (WRR)

É denominado, também, de Escalonamento Baseado em Classe (CBQ - *Class Based Queueing*) ou ainda Escalonamento Customizado (CQ - *Custom Queueing*) [19], [80], [109].

Este algoritmo trabalha com um sistema de alocação absoluta de banda, sendo esta dividida em fatias fixas para cada aplicação, o que garante um percentual da banda a um determinado serviço. A banda reservada para certa aplicação é compartilhada proporcionalmente por todos os usuários. As filas são percorridas de forma cíclica (*round-robin*) [109]. Não havendo a ocupação da banda por alguma fila, a mesma é distribuída entre as demais [28], [85].

De uma maneira geral, seu funcionamento consiste em passar os pacotes inicialmente pelo filtro de tráfego, onde recebem a marcação conforme suas prioridades. Posteriormente são enfileirados nas filas de nível de prioridade correspondente. A cada ciclo do escalonador todas as filas são atendidas, enviando de cada uma delas a quantidade pré-definida de pacotes. Isso é realizado até que seu contador de bytes seja excedido. Contudo, ele respeita o tamanho do pacote, terminando de fazer o envio do último pacote mesmo tendo excedido o contador [28].

Um exemplo de seu funcionamento pode ser observado na Figura 2.4.

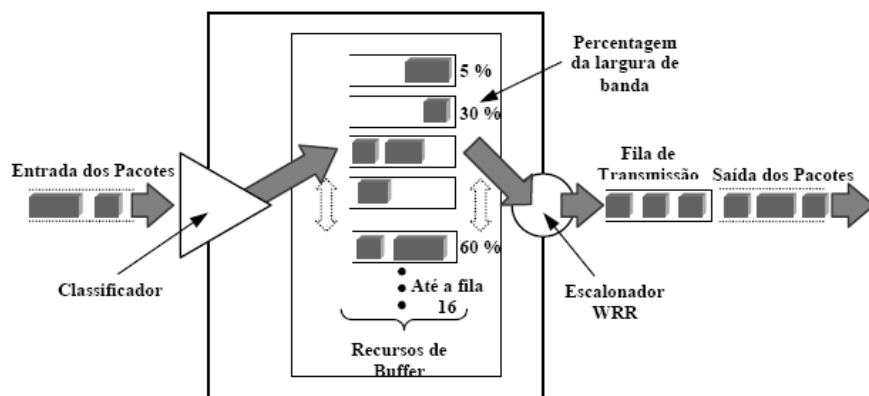


Figura 2.4: Filas Weighted Round Robin. Fonte: [85]

Através da Figura 2.4 pode-se perceber que o sistema possui dezessete (17) filas, sendo que a fila zero (0) é para o atendimento do tráfego do sistema. Por esta razão esta fila tem alta prioridade, sendo a primeira a ser atendida. As outras dezesseis (16) filas são configuradas pelo usuário, devendo este definir quantos bytes serão enviados a cada ciclo de serviço. Vale salientar que esta configuração é estática.

Conforme relatado, as filas são atendidas segundo a política *round-robin*. Além disso, a cada uma das dezesseis (16) filas configuráveis é atribuída uma ponderação, que indicará o percentual de banda que as aplicações daquela classe recebeu. Para cada classe é criada uma fila, as quais são atendidas em ordem decrescente de ponderação, passando para a próxima fila somente após a corrente ficar vazia ou esgotar o percentual de banda reservado a ela. Para isso, um contador de

bytes é associado a cada uma das filas. Este indica a quantidade de bytes enviados a cada ciclo do serviço, possibilitando o controle de banda.

Deste modo, o escalonador WRR controla fluxos com diferenças significativas quanto às necessidades de largura de banda. Assim, em um ciclo de serviço, uma fila com maior largura de banda pode enviar um número superior de pacotes.

O WRR possui limitação, isto é, somente fornecerá uma porcentagem justa de largura de banda para cada fila se todos os pacotes, em todas as filas, forem do mesmo tamanho ou se o tamanho médio dos pacotes for conhecido a priori [64], [109]. Por isso, o escalonador precisa conhecer com antecedência o tamanho médio dos pacotes para que em curtas escalas de tempo seu escalonamento não se torne injusto [85].

2.5.2.4 Escalonador Weighted Fair Queueing (WFQ)

O WFQ é inspirado no algoritmo *Bit-by-bit Round Robin* (BRR), no qual cada fluxo é mantido em uma fila e um bit de cada fluxo é enviado a cada ciclo. No BRR, a banda é dividida igualmente entre os fluxos, havendo igualdade de recursos na utilização do canal. Sua implementação, entretanto, é inviável devido ao envio de apenas um bit, o que causa overhead [42], [85].

O WFQ foi desenvolvido em 1989 por Lixia Zhang, Alan Demers, Srinivasan Keshav e Scott Shenke, sendo que o algoritmo Fair Queueing (FQ) foi proposto por John Nagle em 1987, o qual simula o escalonamento BRR [109].

O WFQ trabalha de maneira análoga ao FQ, possibilitando gerenciar a fila de forma justa, baseando-se na ponderação (peso) de cada fila. Assim, coloca no início da fila o tráfego que possui maior prioridade, de modo a reduzir seu tempo de resposta [114]. Para isso, dada a taxa de bits da porta de saída, o número de filas ativas, a porcentagem de banda alocada para cada fila e o tamanho dos pacotes, o WFQ calcula e atribui um número para cada pacote de entrada. Este número representa a ordem em que os pacotes devem ser escalonados [64], [109].

Assim, o WFQ compartilha a banda de forma homogênea com outros fluxos de menor prioridade, porém enviando uma quantidade menor de pacotes destas filas. Seu funcionamento consiste em classificar o tráfego e dividir em filas de pesos diferentes, sendo os diferentes tipos de tráfego representados na Figura 2.5 como sessões. A banda é compartilhada proporcionalmente ao peso de cada uma das filas, sendo representada na Figura 2.5 pela quantidade de pacotes presentes na esteira de saída. O peso que foi atribuído a cada uma das filas é utilizado para definir a quantidade de bytes a ser enviado de cada um dos fluxos. Para este fim, é associado um contador de bytes a cada uma das filas, sendo transmitidos pacotes até esgotar o contador, então passando para a próxima fila. Caso um pacote não possa ser transmitido devido a seu tamanho, o valor do contador será acumulado para o próximo ciclo do escalonador. Quando não houver mais pacotes na fila antes do saldo estar esgotado, o valor do saldo da fila atual é zerado, iniciando o atendimento da próxima fila [82].

Na Figura 2.5 é apresentado o esquema de funcionamento do WFQ.

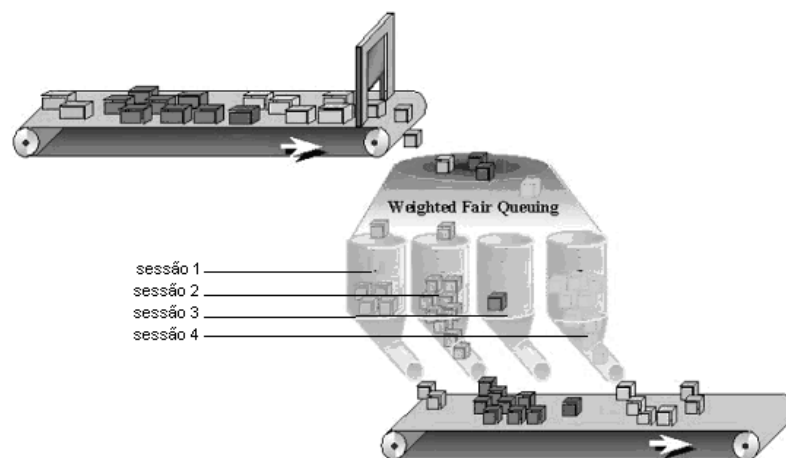


Figura 2.5: Exemplo de Weighted Fair Queuing. Fonte: Adaptada de [113]

Pelo fato de compartilhar toda a largura de banda com os diferentes tráfegos, o escalonador WFQ consegue evitar que as filas cheguem a situações caóticas por falta de recursos. Este mecanismo garante pouco jitter e uma divisão justa de banda entre as diversas aplicações [82]. Em resumo, assegura que nenhuma fila será privada da utilização do canal e que o tráfego receba serviço previsível.

Em geral, para implementações práticas do WFQ, pela sua simplicidade, é utilizado o *Weighted Round Robin* (WRR) [36].

2.5.2.5 Escalonador Deficit Round Robin (DRR)

Proposto por Shreedhar e Varghese [46], [111], o DRR é uma modificação do WRR, porém, não exigindo conhecimento antecipado do tamanho dos pacotes a serem transmitidos [6]. O fluxo é dividido em filas e a cada fila é associada uma quota (valor proporcional ao peso da fila, sendo este expresso em bytes). O atendimento das filas é feito através de um esquema similar ao *round-robin*, onde a cada ciclo são enviados quantos pacotes a quota permitir. Caso um pacote não possa ser enviado por falta de quota, o déficit é acumulado e acrescido à quota do próximo ciclo [39], [59]. Quando não houver buffers disponíveis, faz-se o descarte do último pacote que estiver na fila mais longa.

A Figura 2.6 exemplifica melhor o funcionamento do escalonador DRR.

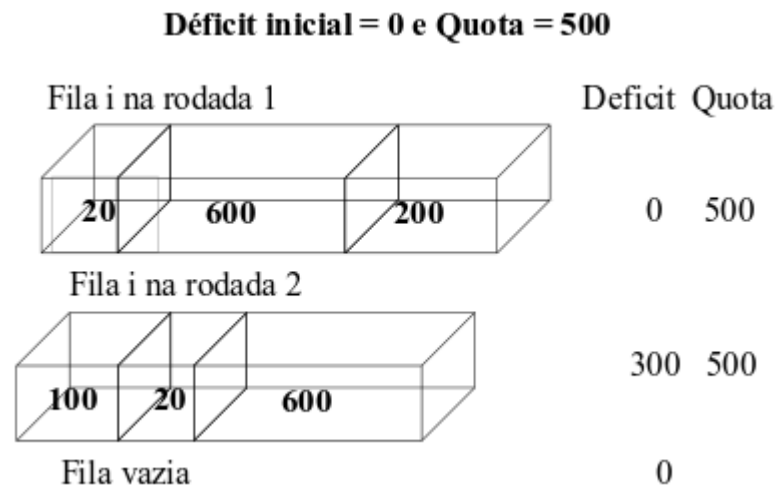


Figura 2.6: Exemplo de Déficit Round Robin. Fonte: [101].

A Figura 2.6 mostra uma fila em um estado inicial com três pacotes de tamanhos 200, 600 e 20, respectivamente. A quota da fila tem peso 500 bytes e o déficit é zero (0), o que significa que o escalonador irá enviar o primeiro pacote, que tem um tamanho de 200 bytes. No momento de

enviar o segundo, de tamanho 600 bytes, não terá quota suficiente. Isso ocasionará um déficit de 300 a ser somado à quota da próxima rodada. Desta forma, na segunda rodada o somatório da quota mais o déficit possibilitará o envio do pacote de tamanho 600, bem como os próximos pacotes de tamanhos 20 e 100 bytes.

Como o objetivo é fazer justiça na distribuição dos recursos, o DRR possui algumas particularidades que o difere. Ele evita a saturação da largura de banda além de suportar as diferentes necessidades de banda dos fluxos, pois observa o tamanho dos pacotes [10], [39]. Além disso, não exige alta complexidade computacional e proporciona equidade na distribuição de largura de banda para pacotes de diferentes tamanhos [6].

Contudo, é importante mencionar que o “DDR não provê limites de retardo rígido,” que é um dos requisitos das aplicações de tempo real [64].

Para a criação de um simulador de eventos discretos para o escalonamento DRR, o mesmo fluxograma apresentado na Figura 2.2 é usado. Contudo, como no escalonador DRR há uma fila para cada um dos tráfegos, o que o difere do escalonador FIFO, apenas as alterações no funcionamento do fluxograma é relatado.

No início do programa há apenas um evento possível: chegada de pacotes. Uma vez que pode ser um pacote de voz, vídeo ou dados verifica-se qual possui o menor tempo de chegada e se inicia o servidor e o processo como um todo.

Depois que o processo já foi iniciado, os dois eventos principais podem ocorrer: a chegada de pacotes ou o término de serviço. No entanto, uma das diferenças no que se refere ao escalonador FIFO é que no DRR há uma variável fila para cada um dos tráfegos.

Se o evento for chegada de pacotes, é identificado que tipo de tráfego deve ser executado (voz, vídeo ou dados), assumindo que o escalonador DRR é cíclico. O próximo passo é a verificação do tamanho do pacote para saber se há quantum suficiente para sua transmissão. Se isso for verdade, o pacote é processado e o valor do quantum é reduzido na mesma proporção que o tamanho do pacote. Uma vez que em cada iteração verifica-se o evento a ser executado, é necessário

identificar o tempo de chegada do próximo pacote. Mediante a existência do pacote, verifica-se o quantum restante referente ao tipo de tráfego em análise. Se o valor do quantum é suficiente, o pacote será tratado. No caso do quantum ser insuficiente, é procurado que tipo de tráfego é o próximo do ciclo e é atribuído ao tempo de chegada o valor da variável *tempo_de_chegada(i+1)*. Após a identificação do pacote atual e do próximo pacote, é analisado o estado do servidor. Se o servidor está livre, mudará para o estado de ocupado e a variável *chegada_pacote* receberá o valor da variável *tempo_de_chegada(i+1)*. No entanto, se o servidor está ocupado, o pacote será armazenado na fila correspondente ou será descartado. Este procedimento é realizado até que todos os pacotes sejam atendidos ou descartados. Esta última situação ocorre apenas em sistemas com buffer finito, e quando todas as posições estão ocupadas.

Já no evento término de serviço é verificado se há pacotes na fila. No caso da fila estar vazia, a variável término de serviço deve receber o valor da variável chegada de pacotes. No caso de existir qualquer pacote na fila, um pacote é atendido, bem como o número de pacotes da fila correspondente é reduzido.

2.5.2.6 Escalonador Baseado em Banda Efetiva

A dificuldade de modelagem do tráfego em rede IP deve-se ao comportamento estatístico dos pacotes de dados e da descoberta de sua natureza fractal [69]. A natureza auto-similar do tráfego de dados torna a modelagem e análise do tráfego muito complexa. Por exemplo, o dimensionamento da capacidade do canal não é apenas o cálculo da largura de banda para curtos intervalos de tempo, sendo que a dependência de longo alcance também deve ser considerada. A maneira de lidar com o tráfego auto-similar é usar o conceito de banda efetiva.

A banda efetiva ou capacidade equivalente é definida como a taxa de serviço mínima para garantir a QoS exigida [63]. Para tanto, a banda efetiva estima uma banda levando em consideração os vários parâmetros de um tráfego, como taxa de pico, taxa média, tempo de atividade, tempo de silêncio, tamanho do buffer, probabilidade de perda de pacotes, entre outros. Uma banda efetiva

calculada nessas condições possibilita uma maneira muito simples de dimensionar enlaces de transmissão [63] ou implementar algoritmos de Controle de Admissão de Conexão (CAC) [45].

No entanto, o cálculo da banda efetiva não é tarefa fácil, dependendo das características das fontes. Algumas expressões para o cálculo da largura de banda efetiva foram derivadas para muitos tipos de tráfego, incluindo periódicos, gaussianos e On/Off [61]. A banda efetiva tem sido também estimada para o tráfego auto-similar utilizando diferentes expressões [63] e [45].

Analisando os algoritmos de escalonamento abordados neste trabalho, o que não provê diferenciação ao tratamento de dados é o escalonamento FIFO, que apenas obedece à ordem de chegada dos pacotes. Contudo, o mesmo é o mais utilizado nos equipamentos de interconexão em função de sua simplicidade.

Os outros escalonadores obedecem a critérios de prioridade estabelecidos para a rede. Mas o que os difere é a maneira como promovem o atendimento às filas de prioridade, podendo servir filas de maior prioridade até que estas estejam vazias, como faz o escalonamento de prioridade ou percorrendo de maneira cíclica e, servindo cada fila através de uma ponderação atribuída antecipadamente.

Independente do modo usado para a priorização, é necessária a distribuição dos recursos entre os usuários ou aplicações. Com este intuito, uma alternativa é utilizar o cálculo da banda efetiva para realizar esta atividade, sendo usado o escalonador DRR para fins de teste. Nessa implementação a quota é proporcional à banda efetiva. Através desse procedimento, pode-se testar a eficiência do escalonamento proposto, que consiste na distribuição da quota por fluxo no escalonador DRR baseando-se na banda efetiva, em relação à utilização da largura de banda e o tempo de espera dos pacotes na fila.

Esta é uma proposta inovadora para um escalonador, sendo que para fins de testes, o cálculo apresentado em [62] será usado para mostrar a eficiência do esquema de escalonamento proposto. O modelo matemático utilizado em [62] é baseado na suposição de que a probabilidade de ocupação do buffer obedece a um decaimento exponencial, apresentado na Equação 1.

$$P\{X > k\} \leq e^{-k\delta} \quad (1)$$

onde k é o tamanho do buffer e δ é uma constante positiva chamada taxa de decaimento assintótico.

Além disso, o modelo é baseado na existência da função

$$h(\delta) = \lim_{t \rightarrow \infty} \frac{\ln E\{\exp(A(t)\delta)\}}{t} \quad (2)$$

onde $A(t)$ é o número de pacotes gerados durante o intervalo de tempo $[0, t]$ e δ é um valor real.

A banda efetiva é dada por

$$c(\delta) = \frac{h(\delta)}{\delta} \quad (3)$$

A avaliação de $h(\delta)$ é em geral difícil. No entanto, para a fonte de fluxo de fluido markoviano (On/Off) as equações acima podem ser resolvidas e a banda efetiva é dada por:

$$c = \alpha + \sqrt{\alpha^2 + \beta^2} \quad (4)$$

onde [93],

$$\alpha = \frac{1}{2\delta} \left(R\delta - \frac{1}{T_{on}} - \frac{1}{T_{off}} \right), \text{ e} \quad (5)$$

$$\beta = \frac{R}{\delta T_{off}} \quad (6)$$

T_{on} é o tempo médio do período ativo e T_{off} é o tempo médio do intervalo de silêncio. R é a taxa de geração de pacotes no período ativo.

O valor de δ é dado por

$$\delta = \frac{\ln(1/\varepsilon)}{k}, \quad (7)$$

onde ε é a taxa de perda desejada.

Pode-se perceber na Equação 4 que para a realização do cálculo é necessário conhecer algumas características do tráfego que circula na rede em análise, sendo este do tipo On/Off. Este modelo é definido pela alternância entre dois períodos: On e Off. No período On, os pacotes são gerados a uma taxa de pico (rajada), ou seja, a fonte está ativa. Por sua vez, o período Off consiste em um intervalo de silêncio, não ocorrendo emissão de nenhum pacote.

Um exemplo de cálculo de banda efetiva para fontes auto-similares considerando uma e múltiplas fontes é apresentado em [63].

Para testar o algoritmo de escalonamento proposto, um simulador de eventos discretos foi desenvolvido, baseando-se no fluxograma apresentado na Figura 2.2. O fluxograma funciona de acordo com o escalonador DRR, sendo que a distribuição da quota é feita usando o cálculo da banda efetiva proposta em [62].

2.6 CONCLUSÕES DO CAPÍTULO

Neste capítulo foram apresentados os conceitos e as soluções para redes convergentes, bem como definições para escalonadores de tráfego e alguns algoritmos de escalonamento para estas redes.

A partir do estudo realizado, constatou-se que não há como idealizar uma nova tecnologia de rede ou protocolos de roteamento e endereçamento sem que a mesma deixe de suportar a transmissão das mais diversas mídias, sobretudo voz, vídeo e dados, sendo aqui denominadas de multimídia. Neste sentido, tendo em vista a disseminação e o uso massivo da Internet, uma das soluções mais viáveis como plataforma de convergência é a Rede IP.

No entanto, a rede IP não foi projetada visando atender a Qualidade de Serviço (QoS) dos tráfegos multimídia. Deste modo, uma das soluções propostas foi apresentada pela *Internet Engineering Task Force* (IETF): o DiffServ [14]. O DiffServ propõe uma diferenciação dos vários fluxos de tráfegos em classes e cada classe é tratada de uma mesma maneira em cada um dos nós da rede. Entretanto, o DiffServ não propõe nenhum tipo de escalonador para tratamento desses tráfegos.

Para controlar a QoS dos tráfegos e os recursos da rede é necessário um esquema eficiente de escalonamento de tráfego. Embora o algoritmo FIFO tenha sido projetado para atender apenas um tipo de tráfego, o de dados, ele é o mais comumente usado. Assim o mesmo pode não ser apropriado para atender a uma variedade de tráfegos. Deste modo, seu desempenho é um dos focos do presente trabalho.

Por sua vez, outros algoritmos de escalonamento de tráfego foram propostos na literatura que tentam solucionar, principalmente, o problema de equidade dos tráfegos tais como o *Weighted Round Robin* (WRR), *Weighted Fair Queueing* (WFQ) [33] e *Deficit Round Robin* (DRR) [111], [118]. O escalonador *Priority Queue* (PQ) também é uma opção. No entanto, conforme relatado, ele não permite a distribuição justa entre os fluxos. Além disso, o mesmo pode e deve ser usado com outro escalonador, sendo que o atendimento de cada fila pode seguir o esquema FIFO.

Para confrontar os resultados obtidos com o escalonador FIFO, um escalonador que permite distribuir os recursos com equidade também é analisado, optando-se pelo escalonador DRR. O escalonador DRR foi escolhido por ser um algoritmo bem difundido e já consolidado na literatura, por sua simplicidade computacional, por permitir distribuir quotas diferenciadas para os diversos tipos de tráfego, atendendo desta forma a QoS requerida pelos usuários e aplicações. Além disso, há facilidade de modificar este algoritmo de escalonamento, de modo a obter bons resultados no que se refere ao atraso e perda de pacotes, como pode ser comprovado em [26] e [102].

Entretanto, a procura por um escalonador adequado para as redes multimídias continua a ser um campo de estudo.

3 MODELOS DE FONTES DE TRÁFEGO PARA REDES CONVERGENTES

Para estudar o comportamento dos escalonadores em redes convergentes é necessário escolher modelos convenientes de fontes de tráfego que representem com maior aproximação as fontes reais utilizadas em redes convergentes.

A modelagem de fontes de tráfego objetiva a criação de modelos matemáticos que possibilitem formar um padrão para cada fonte [88]. Uma fonte de tráfego não precisa ser sofisticada [53]. Deve possibilitar a modelagem adequada dos dados, encontrar um modelo apropriado para a geração de pacotes de dados, adequar-se aos diferentes tipos de tráfego, atingir os objetivos estabelecidos e garantir a qualidade de serviço fornecida ao usuário.

Neste capítulo, são apresentados alguns modelos de fontes de tráfego existentes e já referendados na literatura, relatando os motivos para sua utilização. O objetivo é identificar o mais usado, que seja de fácil implementação. Será dado destaque para a apresentação de propostas de fontes baseadas no modelo On/Off, os impactos causados usando os modelos propostos, bem como demonstrando onde é mais adequada a utilização de cada uma das fontes.

3.1 Modelo de Poisson

O modelo de Poisson é um dos modelos mais antigos de tráfego e foi utilizado pela primeira vez no dimensionamento de enlaces telefônicos [47].

O processo de chegadas poissonianas é caracterizado como um processo de renovação, cujo tempo entre chegadas é exponencialmente distribuído. É um processo markoviano particular em que o número de chegadas em certo intervalo de tempo obedece a uma distribuição de Poisson.

O modelo de chegadas poissonias foi muito utilizado nos dimensionamentos de enlaces e nos estudos de filas em redes de computadores. Entretanto, devido ao aumento de volume de dados e na descoberta da natureza auto-similar dos tráfegos em redes de computadores, esse modelo está sendo substituído por modelos mais complexos.

O modelo de Poisson possui algumas limitações como na captura da correlação das chegadas e no crescimento abrupto do tráfego, o que influencia negativamente o desempenho em filas [127]. Porém, tem a vantagem de ser analiticamente mais simples, e pode ainda ser utilizado em alguns casos de dimensionamento de redes.

3.2 Modelo On/Off

O modelo de tráfego On/Off é um dos mais comuns para o estudo dos sistemas de redes [16]. Ele possui períodos de atividade, no qual os pacotes são enviados e períodos de silêncio onde não há transmissão de dados.

Existem ainda variações do modelo On/Off, sendo algumas delas descritas a seguir.

3.2.1 Modelo On/Off de Pareto

Neste modelo, os estados de atividade e inatividade de um tráfego de redes são distribuídos conforme a distribuição de Pareto [127].

Vários autores emitiram parecer sobre a distribuição de Pareto com enfoques específicos. Apresenta-se um resumo:

- tem uma forma de distribuição acumulativa;

- produz tempos de chegadas idênticos e independentes [24] (p. 04);
- é aplicada a modelos de tráfegos auto-similares [24] (p. 04);
- tem variância infinita [127] (p. 37);
- pode resultar em um tráfego auto-similar [20];
- apresenta cauda pesada [43] (p. 04);
- se utilizado um número elevado mas não infinito de fontes, o tráfego resultante é uma aproximação de um processo puramente auto-similar [20].

O modelo On/Off de Pareto, da classe das distribuições de cauda-pesada é a mais apropriada para modelagem de tráfego em muitos casos. É um modelo bastante adequado para tráfegos com fluxos de *Long-Range Dependent* (LRD - Dependência de Longo Alcance), podendo ser ajustado a um rastro de medida [15] (p. 02).

3.2.2 Modelo On/Off Exponencial

As fontes de tráfego On/Off Exponenciais alternam entre períodos ativos de emissão de dados (períodos de On) e períodos inativos (períodos de Off). Durante o período de On, os pacotes ou dados são gerados a uma taxa constante (caracterizando uma rajada). No tempo de Off nenhum dado ou pacote é gerado. Esses períodos são caracterizados por uma média ou tempo de interchegada modelados por uma distribuição exponencial. O modelo On/Off Exponencial é bastante usado para modelar tráfego de voz, e é capaz de reproduzir correlações e explosividade de um tráfego real [103] (p. 72).

O modelo On/Off é usado geralmente quando há a necessidade de emular o comportamento e dimensionamento do tráfego de rede. Por exemplo, análises de estruturas de tráfegos IP são realizadas predominantemente através de modelos On/Off [24] (p. 05).

Com períodos On/Off exponencialmente distribuídos, a fonte pode ser modelada como uma cadeia de Markov de dois estados, conforme apresentada na Figura 3.1.

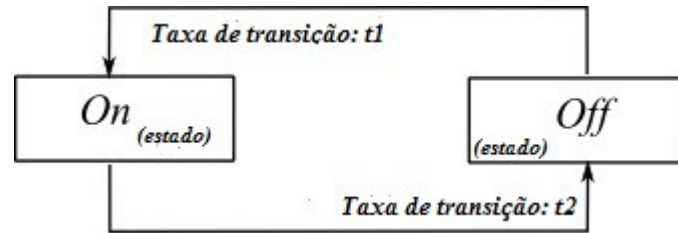


Figura 3.1: Modelo On/Off simples com taxas de transição t_1 e t_2 . (Adaptada de [24], (p. 05))

3.3 Modelo Auto-Similar

Auto-similaridade ou auto-similaridade estatística implica que as flutuações das taxas de tráfego mostram um padrão similar durante muitas escalas de tempo. Outra forma de definir auto-similaridade é através de LRD, significando que a correlação deteriora mais lentamente do que nos modelos convencionais de tráfego. Estes fenômenos não são mutuamente independentes [105].

Devido à complexidade matemática, é muito difícil a modelagem das fontes auto-similares. Alguns modelos utilizaram o *fractional Brownian Motion* (fBM) [105]. Contudo, como o comportamento fractal observado é limitado a algumas escalas de tempo, modelos pseudo auto-similares foram propostos.

Outros modelos utilizaram a distribuição de Pareto como a fonte On/Off de Pareto apresentada na subseção 3.2.1.

Através dos argumentos apresentados em [38], [41], [56], [105], [127], [130], verifica-se que não há consenso sobre a utilização do modelo auto-similar para os diferentes tráfegos. Embora as ressalvas não sejam muitas, a estrutura do tráfego da Internet é um fenômeno muito complexo e necessita de aprofundamento da análise.

3.4 JUSTIFICATIVA PARA USAR FONTES ON/OFF E PARÂMETROS PARA A SUA GERAÇÃO

O tráfego poissoniano não é mais conveniente, depois que W. E. Leland *et al* [69] mostraram em seu estudo a natureza auto-similar do tráfego de dados. Desde então, muitos tipos de tráfego auto-similares e modelos de fila têm sido propostos [44], [75] e [91]. No entanto, tais modelos matemáticos de tráfego são complexos e ainda não são totalmente adequados aos tráfegos reais.

Além disso, a escolha dos modelos de tráfego para uma rede com tráfego multimídia, contendo voz, vídeo e dados é uma tarefa muito difícil, porque cada tráfego tem características diferentes, ficando isso evidente com o surgimento das redes convergentes. Contudo, não é fácil fazer a modelagem de modo a representar todas as variações e características. Em geral, o tipo de tráfego auto-similar é modelado como a agregação de tráfegos entrantes em um nó, o que pode gerar perda de características de tráfegos específicos. Para modelar uma fonte individual, muitas abordagens diferentes foram adotadas. Uma das abordagens utilizadas em [1] consiste em dividir uma conexão com a Internet em seções, cada uma das seções com diferentes distribuições, o que torna o processo relativamente complexo de implementar.

O modelo de tráfego On/Off para cada fonte é muito simples de implementar e, além disso, o agregado dessas fontes representa com boa aproximação o tráfego auto-similar e, dessa forma, tem sido utilizado para o estudo do desempenho das redes em [57] e [100].

O modelo de tráfego On/Off com distribuição exponencial foi a opção para modelar todos os tipos de fontes individuais, inclusive para dados, devido à sua simplicidade. Afinal, os modelos de fontes HTTP e auto-similares são bem mais complexos, sendo difíceis de modelar e implementar.

Além disso, a agregação de fontes individuais On/Off resulta em tráfego auto-similar [83].

Uma comparação entre tráfegos On/Off Exponencial e a *Constant Bit Rate* (CBR) foi realizada em [99]. Os autores descrevem que CBR é bom para aplicações de dados, mas não é adequado para aplicações multimídia. Por outro lado, o tráfego exponencial pode representar as características de

tráfegos multimídia, pois possui variação na taxa de geração de pacotes de dados, seguido por um período Off. Atualmente os autores estão empenhados em completar os protocolos atuais de roteamento para aplicações multimídia usando fontes de tráfego exponencial.

Deste modo, com base nas razões apresentadas, o modelo On/Off foi usado para os estudos. Para a realização das simulações referentes às fontes de tráfego, foram utilizados os parâmetros apresentados na Tabela 3.1, os quais são valores mencionados na literatura.

Tabela 3.1: Taxa de pico, tamanho dos pacotes e média dos intervalos *Off* das fontes de dados, voz e vídeo.

Tipos de Tráfegos	Tamanho de Pacote (bits)	Taxa de Pico (bps)	Tempo de Off (s)
Voz	1360	65536	0,0016
Vídeo	4096	1048576	0,001
Dados	4096	307200	0,003

Alguns parâmetros apresentados na Tabela 3.1 são calculados por meio de valores aleatórios com distribuição exponencial negativa, como o tamanho dos pacotes, o tempo dos intervalos On e Off e a quantidade de pacotes gerados por intervalo. Para isso, utiliza-se a equação (8) [115].

$$X = -\frac{\ln(U)}{\lambda} \quad (8)$$

Na equação (8), U representa um número aleatório com distribuição uniforme no intervalo (0;1]. Esse valor é obtido através da função “rand” da maioria das linguagens de programação. E, λ representa a taxa média de geração de pacotes.

3.5 DESCRIÇÃO DAS FONTES ON/OFF DESENVOLVIDAS

Definidos os parâmetros para a geração de tráfegos de voz, vídeo e dados, e justificadas as razões para que isso seja realizado usando o modelo de fonte On/Off Exponencial, três fontes foram implementadas na linguagem de programação C++ Builder.

Neste intuito, este tópico apresenta o funcionamento das fontes geradas, bem como algumas conclusões sobre o impacto que os parâmetros e a forma de implementação podem ocasionar na utilização dos recursos computacionais.

3.5.1 Descrição da Fonte On/Off Fixa

A fonte On/Off Fixa gera apenas um pacote a cada Intervalo On (Ton). Além disso, os pacotes gerados possuem tamanho fixo e os intervalos de On também são fixos, sendo que seu tempo de duração é calculado dividindo-se o tamanho do pacote pela taxa de pico da fonte em questão.

Então, em uma simulação utilizando o tráfego de voz, com tamanho do pacote fixo de 1360 bits e taxa de pico com valor de 65536 bits por segundo (bps), o intervalo ativo terá duração de 0.02075 s.

Após determinar o intervalo de On é calculado o tempo de chegada de cada pacote. Este é obtido somando-se o tempo de chegada dos pacotes anteriores acrescido dos intervalos de silêncio e a soma do tempo de chegada do pacote atual. Logo, em um tráfego de voz, o tempo de chegada do primeiro pacote será 0.02075 s, ou seja, zero mais seu próprio tempo de chegada.

No entanto é importante mencionar que o tempo de silêncio não é fixo. Ele possui distribuição exponencial negativa, sendo calculado conforme a equação 8.

Neste modelo de geração de fonte os primeiros vinte por cento dos pacotes gerados são descartados para fins estatísticos. Este procedimento é adotado para que o sistema entre em estabilidade.

Depois que todos os pacotes são gerados, os resultados são armazenados em arquivo de texto. Na Figura 3.2 é apresentada a imagem de uma parte deste arquivo.

```

99996; 1360.000000; 2237.443604; 0.000543
99997; 1360.000000; 2237.465332; 0.000933
99998; 1360.000000; 2237.486816; 0.000742
99999; 1360.000000; 2237.507812; 0.000287
Quant. Intervalos ON: 80000;
Totalizador de ON: 1660.156250;
Quant. Intervalos OFF: 80000;
Totalizador de OFF: 129.707642;
Media de ON: 0.020752;
Media de OFF: 0.001621

```

Figura 3.2: Arquivo de texto gerado pelo programa Fonte On/Off Fixa

No arquivo, apresentado em parte pela Figura 3.2, nas primeiras quatro linhas constam os dados referentes ao tráfego gerado, seguido de dados estatísticos. Assim, na primeira consta o índice do pacote gerado, seguido do tamanho do pacote que é fixo (1360 porque é um tráfego de voz). O tempo de chegada do pacote é apresentado na terceira coluna e, por último, o intervalo de Off que segue uma distribuição exponencial.

É importante mencionar que a Figura 3.2 foi extraída a partir da simulação de um tráfego de 100000 pacotes de voz. Como existe um descarte dos primeiros vinte por cento de pacotes gerados, então só são contabilizados 80000 intervalos de On e de Off. O totalizador de On representa a soma de todos os intervalos On, ou seja, o somatório do tamanho dos pacotes dividido pela taxa de pico da fonte em questão, e o totalizador de Off consiste na soma de todos os intervalos Off, ambos considerando o descarte dos pacotes. A média de On representa o total de On dividido pela quantidade de intervalos On, e a média de Off, o total de Off dividido pela quantidade de intervalos Off, ambos considerando o descarte estatístico.

3.5.1.1 Análise dos Resultados da Fonte On/Off Fixa

Para analisar os resultados do programa Fonte On/Off Fixa foram gerados 100000, 200000 e 300000 pacotes para cada um dos tráfegos de voz, de vídeo e de dados, sendo os resultados gerados dispostos na Tabela 3.2.

Tabela 3.2: Resultado das fontes geradas pelo programa Fonte On/Off Fixa

	Voz			Vídeo			Dados		
Qtd.de Pacotes	100000	200000	300000	100000	200000	300000	100000	200000	300000
Qtd. Intervalos On	80000	160000	240000	80000	160000	240000	80000	160000	240000
Total de On	1660,1563	3320,3125	4970,0635	312,5000	625,0000	937,5000	1065,6091	2130,8159	3205,0347
Qtd. Intervalos Off	80000	160000	240000	80000	160000	240000	80000	160000	240000
Total de Off	129,7532	259,5222	390,1918	81,2252	162,0805	243,3335	243,2891	485,8478	730,7843
Média de On	0,0208	0,0208	0,0207	0,0039	0,0039	0,0039	0,0133	0,0133	0,0134
Média de Off	0,0016	0,0016	0,0016	0,0010	0,0010	0,0010	0,0030	0,0030	0,0030

Através da Tabela 3.2 pode-se verificar que independente da quantidade de pacotes gerada, vinte por cento está sendo descartada. Além disso, se for dividido o tamanho do pacote pela taxa de pico da fonte em análise e o valor encontrado for multiplicado pelo número de pacotes considerados para fins estatísticos, o resultado é exatamente equivalente ao Total de On informado na Tabela 3.2. Consoante ao Total de Off não há como verificar se este valor está correto. Contudo, ele é usado para o cálculo da Média de Off. Então, se este não estiver certo, a Média de Off apresentará problemas. Já a Média de On está convergindo diretamente para o que foi especificado na Tabela 3.1, tendo pequena alteração na quarta casa decimal nos tráfegos de voz e dados. Isso pode ser comprovado, pois além de corresponder ao valor mencionado na Tabela 3.1, se for dividido o Total de On pela Qtd. de Pacotes mencionada na Tabela 3.2, o mesmo valor será obtido.

É importante relatar que os dados apresentados na Tabela 3.2 mostram apenas quatro casas após a vírgula. Assim se não houvesse delimitação no número de casas após a vírgula, nas simulações com tráfego de voz, mesmo com a diferença de geração de 100000 pacotes em cada simulação, a média de On variou apenas 0.0001 s em relação a menor e a maior média. Isso comprova que os valores estão convergindo.

No que tange a média de Off, em todos os tipos de tráfego gerados, os resultados obtidos convergem exatamente para as médias descritas na Tabela 3.1, sem nenhuma variação até a quarta casa decimal.

3.5.2 Descrição da Fonte On/Off Variável 1.0

Na Fonte On/Off Variável 1.0, aqui denominada como Fonte 1.0, a quantidade de pacotes gerados por intervalo pode ser fixa ou não, dependendo da opção do usuário. Caso o usuário opte por gerar uma quantidade variável de pacotes por intervalo, então, em um único intervalo ativo pode ser gerado mais de um pacote do tipo de tráfego escolhido. Esta quantidade segue a distribuição exponencial negativa. Além disso, o tamanho dos pacotes também não é fixo, sendo gerado com distribuição exponencial negativa.

Para a geração dos valores referentes ao tamanho do pacote, a taxa de pico e a média dos intervalos de Off, há necessidade que o usuário inclua as informações referentes aos valores médios destes parâmetros, sendo usados os valores apresentados na Tabela 3.1.

Desta forma, caso o usuário opte por gerar uma quantidade de pacotes variável por intervalo ativo, o valor médio deve ser definido pelo usuário, sendo este gerado com distribuição exponencial. Além disso, do valor exponencial gerado, será retornado apenas sua parte inteira, pois não há pacotes fracionados. No entanto, caso o usuário tenha selecionado a opção para gerar uma quantidade fixa de pacotes por intervalo, então a quantidade de pacotes definida será gerada em todos os intervalos ativos.

Após determinar a quantidade de pacotes do intervalo ativo atual, será gerado o tamanho de cada pacote, sendo que apenas a parte inteira do valor exponencial gerado será considerada.

O tempo de chegada de cada pacote é calculado dividindo o seu tamanho pela taxa de pico do tráfego em questão, acrescido dos demais tempos dos intervalos On e Off.

Por exemplo: supondo que em uma simulação com tráfego de voz foi gerado apenas um pacote no intervalo anterior com tamanho de 1375 bits. Logo, o intervalo On será o tamanho do pacote dividido pela taxa de pico, ou seja, 0.02098 s, o que converge para os valores apresentados na Tabela 3.1. Supondo que depois foi gerado um intervalo Off de 0.00163 s. Então, se o pacote atual possui o tamanho de 1350 bits, seu intervalo On será de 0.02059 s e seu tempo de chegada será de 0,04320 s, equivalente a soma dos valores 0.02098, 0.00163, 0.02059.

Os intervalos Off também são gerados com distribuição exponencial negativa usando a equação 8, sendo descartados vinte por cento dos pacotes gerados para fins estatísticos.

Após a geração de todos os pacotes, o programa Fonte On/Off Variável 1.0 gera um arquivo contendo as principais informações relativas a esta fonte. A Figura 3.3 mostra uma parte do arquivo gerado.

```
99996; 4; 1416.000000; 0.021606; 2140.225830; 0.000682
99997; 1; 402.000000; 0.006134; 2140.232422; 0.001049
99998; 1; 3826.000000; 0.058380; 2140.291748; 0.001869
99999; 2; 1952.000000; 0.029785; 2140.323486
Quant. Intervalos ON: 31659;
Totalizador de ON: 1656.880981;
Quant. Intervalos OFF: 31658;
Totalizador de OFF: 50.904457;
Média do tamanho dos pacotes: 1357.316900;
Média de pacotes por intervalo: 2.526928;
Média de ON: 0.052335;
Média de OFF: 0.001608
```

Figura 3.3: Arquivo gerado pelo programa Fonte On/Off Variável 1.0

Na Figura 3.3, a primeira coluna representa o índice de geração dos pacotes, sendo que o último valor, apresentado na quarta linha, ou seja, 99999 é o último pacote gerado. Afinal, foram gerados cem mil pacotes e o índice começa em zero.

A segunda coluna refere-se à quantidade de pacotes que foram gerados no intervalo On. Assim, se três pacotes foram gerados no mesmo intervalo, o número três aparecerá em três linhas consecutivas. Por exemplo, como na primeira linha desta coluna aparece o valor quatro, representa que neste intervalo foram gerados quatro pacotes, sendo que este número se repetiu nas linhas 99993, 99994, 99995 e 99996.

Na terceira coluna consta o valor referente ao tamanho do pacote, seguido do intervalo de On e do tempo de chegada. Por fim, o intervalo Off é apresentado na sexta coluna. Como este é gerado apenas uma vez a cada intervalo On, há algumas linhas que não contêm esta informação.

No final do arquivo da Figura 3.3 estão as informações estatísticas, desconsiderando os primeiros vinte por cento dos pacotes gerados no programa Fonte 1.0.

3.5.2.1 Análise dos Resultados da Fonte On/Off Variável 1.0

Para analisar os resultados gerados pelo programa Fonte On/Off Variável 1.0, foram feitas simulações com 100000, 200000 e 300000 pacotes de cada um dos tráfegos apresentados na Tabela 3.1, e seus respectivos parâmetros. Os pacotes foram gerados com quantidade fixa e média de dois pacotes por intervalo, sendo que os resultados obtidos são dispostos na Tabela 3.3.

Tabela 3.3: Resultados das fontes geradas pelo programa Fonte On/Off Variável 1.0

Qtd.Pacotes por Intervalo	Voz			Vídeo			Dados		
	Quantidade Fixa								
	2			2			2		
Qtd.de Pacotes	100000	200000	300000	100000	200000	300000	100000	200000	300000
Qtd. Intervalos On	40000	80000	120000	40000	80000	120000	40000	80000	120000
Total de On	1655,5748	3326,0551	4989,4541	313,6692	626,21613	939,89803	1069,6341	2141,7889	3210,2498
Qtd. Intervalos Off	39999	79999	119999	39999	79999	119999	39999	79999	119999
Total de Off	63,8760	128,5727	192,7059	40,3114	79,9015	120,3745	120,1931	240,5711	361,7777
Média do tamanho dos pacotes	1356	1362	1362	4111	4104	4106	4107	4112	4109
Média de pacotes por intervalo	2,0000	2,0000	2,0000	2,0000	2,0000	2,0000	2,0000	2,0000	2,0000
Média de On	0,0414	0,0416	0,0416	0,0078	0,0078	0,0078	0,0267	0,0268	0,0268
Média de Off	0,0016	0,0016	0,0016	0,0010	0,0010	0,0010	0,0030	0,0030	0,0030

Qtd.Pacotes por Intervalo	Quantidade Média								
	2			2			2		
	100000	200000	300000	100000	200000	300000	100000	200000	300000
Qtd. Intervalos On	31646	63152,667	94942	31591	63381	95184	31625,667	63281	94645
Total de On	1660,3816	3331,2805	4972,5788	313,16419	625,91974	941,42155	1067,7805	2141,1663	3206,2437
Qtd. Intervalos Off	31645	63152	94942	31591	63381	95183	31625	63281	94645
Total de Off	50,7826	101,2686	152,3986	31,7054	63,7806	95,2592	95,2478	190,7243	284,8602
Média do tamanho dos pacotes	1360	1364	1358	4105	4102	4113	4100	4111	4104
Média de pacotes por intervalo	2,5280	2,5335	2,5279	2,5325	2,5244	2,5214	2,5296	2,5284	2,5358
Média de On	0,0525	0,0527	0,0524	0,0099	0,0099	0,0099	0,0338	0,0338	0,0339
Média de Off	0,0016	0,0016	0,0016	0,0010	0,0010	0,0010	0,0030	0,0030	0,0030

Pode-se perceber, através da Tabela 3.3, que o programa Fonte On/Off Variável 1.0 está funcionando corretamente. Esta afirmação pode ser comprovada pelo fato de que em todas as simulações, os primeiros vinte por cento dos pacotes foram descartados. Isso pode ser observado, pois quando foram gerados dois pacotes em cada intervalo ativo (Fonte On/Off Variável 1.0 com Quantidade Fixa), o número de intervalos On é exatamente a metade da quantidade de pacotes considerados para fins estatísticos nas simulações.

Na geração com quantidade fixa de dois pacotes, o tempo médio de Off corresponde exatamente ao que foi definido na Tabela 3.1. Isso independente do tipo de tráfego analisado, o que demonstra que a quantidade de intervalos Off e o Total de Off estão sendo calculados corretamente,

uma vez que dividindo o Total de Off pela quantidade de intervalos Off, obtém-se a Média de Off. O mesmo é válido para a Média de On.

No que se refere a Média de On no programa Fonte On/Off Variável 1.0 com Quantidade Fixa, se for multiplicada a quantidade de pacotes do intervalo pelo tamanho do pacote definido, e o valor obtido for dividido pela taxa de pico da fonte em análise, também será obtida a Média de On, o que comprova o funcionamento adequado do programa.

Consoante ao tamanho dos pacotes, nas simulações com o programa Fonte 1.0 com Quantidade Fixa, percebe-se que os valores gerados convergem para os especificados na Tabela 3.1. A maior diferença é produzida no tráfego de vídeo e não chega a dois por cento.

Considerando a aplicação Fonte On/Off Variável 1.0 com Quantidade Média, como o número médio de pacotes por intervalo é superior, o total de intervalos On passa a ser menor. Contudo, a média de On está correta, tendo em vista que dividindo o Total de On pela Quantidade de Intervalos On, a média apresentada é obtida. O mesmo ocorre com a média de Off, sendo que o resultado obtido é o mesmo apresentado na Tabela 3.1 para todos os tipos de tráfegos.

No que se refere ao tamanho dos pacotes, tanto com a Fonte On/Off Variável 1.0 com Quantidade Fixa ou Média, o mesmo comportamento acontece, uma vez que os resultados são similares e convergem para os dados especificados na Tabela 3.1. Porém, quanto a quantidade de pacotes gerados por intervalo, embora os resultados convirjam para o que foi definido pelo usuário, há uma variação de aproximadamente 0,5 pacotes nos três tipos de tráfegos em análise.

3.5.3 Descrição da Fonte On/Off Variável 2.0

A Fonte On/Off Variável 2.0, chamada neste trabalho de Fonte 2.0, é similar a Fonte 1.0, pois o tempo de On, o tamanho dos pacotes e a quantidade de pacotes por intervalo são gerados com distribuição exponencial negativa. Contudo, a Fonte 2.0 difere do modelo anterior pelo fato de primeiro gerar o intervalo de On e após um pacote. Caso seja possível enviar o pacote no intervalo

ativo atual, isso é realizado. Caso contrário, o pacote é armazenado para ser enviado no próximo intervalo. Vale salientar que não há acúmulo do tempo ativo caso o pacote não seja enviado neste intervalo. O que acontece é que o tempo de On não utilizado é considerado como tempo de Off.

Este programa inicia com o intervalo On, seguido de um intervalo Off gerado também com distribuição exponencial negativa. Os intervalos de On são gerados tendo como base o tempo necessário para gerar em média a quantidade de pacotes definida pelo usuário na configuração do programa. Quanto ao tamanho dos pacotes, estes são gerados com distribuição exponencial negativa, considerando apenas a parte inteira do valor gerado e o valor médio informado pelo usuário.

Quando um pacote é gerado, o programa verifica se o tempo de On é suficiente para enviá-lo. Caso não seja, este pacote é armazenado para envio no próximo intervalo On e o tempo de On que sobrou deste intervalo é considerado tempo de Off. Deste modo, antes de gerar um pacote é necessário verificar se não há algum armazenado. Se houver, o pacote será transmitido. Caso contrário, um novo pacote será gerado. Contudo, se não houver pacote armazenado e o pacote gerado tiver um tempo de processamento maior que o intervalo de On, serão gerados novos intervalos On até que seja possível enviar este novo pacote. Este procedimento foi feito para evitar que pacotes gerados sejam desconsiderados. Além disso, não pode haver um intervalo On sem que tenha ocorrido atividade, bem como após um intervalo Off é obrigatório que haja um intervalo On. Assim, pode ocorrer que sejam desconsiderados vários intervalos On, até que se obtenha um intervalo em que um pacote seja transmitido. Por esta razão, mesmo que o intervalo On possua distribuição exponencial negativa, sua média não irá convergir tão bem quanto no programa Fonte On/Off Variável 1.0.

Após gerar o tempo do intervalo On o programa gera a quantidade de pacotes a ser transmitida neste intervalo, sendo que a cada pacote gerado, seu tempo de chegada é calculado.

O cálculo do tempo de chegada dos pacotes é feito somando-se o total de intervalos On, ao total de intervalos Off e ao valor excedente do intervalo On. Desta forma, o tempo de chegada do primeiro pacote será o próprio tamanho do pacote dividido pela taxa de pico do tráfego em análise,

tendo em vista que o total de intervalos On e Off ainda estão zerados. Já o tempo de chegada do segundo pacote se este for gerado no mesmo intervalo que o pacote anterior, será o tempo de chegada do pacote anterior acrescido do seu próprio tempo de chegada. Caso o segundo pacote seja gerado no próximo intervalo On, seu tempo de chegada será o somatório do tempo de chegada do pacote anterior, mais o tempo do intervalo de Off, adicionado ao seu próprio tempo de chegada.

Quanto aos intervalos de Off, estes também são gerados com distribuição exponencial negativa e os cálculos das variáveis estatísticas desconsideram os primeiros vinte por cento dos pacotes gerados.

Após a geração de todos os pacotes, o programa Fonte 2.0 armazena os principais dados dos pacotes gerados em um arquivo texto, sendo este similar ao do programa Fonte On/Off Variável 1.0.

3.5.3.1 Análise dos Resultados da Fonte On/Off Variável 2.0

100000, 200000 e 300000 pacotes foram gerados para os tráfegos de voz, vídeo e dados com quantidade média de 2 pacotes por intervalo, para a análise dos resultados do programa Fonte On/Off Variável 2.0.

Assim, na Tabela 3.4 constam os dados estatísticos referentes aos tráfegos gerados.

Tabela 3.4: Resultados das fontes geradas pelo programa Fonte On/Off Variável 2.0

	Voz			Vídeo			Dados		
Qtd.de Pacotes	100000	200000	300000	100000	200000	300000	100000	200000	300000
Qtd. Intervalos On	26966	53867	80678	26848	54003	80825	27046	54159	80807
Total de On	1663,4746	3323,4344	4994,2069	313,21953	627,19948	939,42356	1069,0328	2140,7884	3202,946
Qtd. Intervalos Off	26965	53866	80677	26846	54001	80823	27045	54158	80805
Total de Off	460,1273	916,7866	1375,1652	123,5181	248,1649	371,5076	402,4160	806,8777	1200,4021
Média do tamanho dos pacotes	1363	1361	1364	4105	4110	4104	4105	4110	4100
Média de pacotes por intervalo	2,9667	2,9703	2,9748	2,9798	2,9628	2,9694	2,9580	2,9543	2,9701
Média de On	0,0617	0,0617	0,0619	0,0117	0,0116	0,0116	0,0395	0,0395	0,0396
Média de Off	0,0171	0,0170	0,0170	0,0046	0,0046	0,0046	0,0146	0,0149	0,0149

Analisando os dados apresentados na Tabela 3.4 pode-se observar que se for dividido o total de On pela quantidade de intervalos On, a média de On será obtida. O mesmo ocorre no que se

refere à média de Off, o que demonstra que os resultados estão corretos. Contudo, pode-se perceber que a média de Off não corresponde fidedignamente aos valores apresentados na Tabela 3.1, sendo que os valores apresentados na Tabela 3.4 são um pouco superiores. Isso ocorre pelo fato que o tempo de On excedente em função de não ter sido possível a transmissão de um pacote é adicionado ao tempo de Off. Deste modo, isso já era esperado. Porém este aumento não é muito significativo.

Também é possível verificar que o aumento de 100000 pacotes em cada simulação não gera diferenças expressivas nas médias de On, sendo a maior diferença ocorrida no tráfego de voz, correspondendo a 0.0002 s. Isso comprova que com 100000 pacotes o programa já convergiu. Este fato também é verdade para as médias de Off, na qual a maior diferença é de 0.0003 s no tráfego de dados.

Consoante ao tamanho médio dos pacotes, os valores obtidos são muito próximos aos apresentados na Tabela 3.1, demonstrando que há convergência.

O único valor que não está convergindo diretamente para o valor informado é a quantidade média de pacotes por intervalo. O valor médio informado foi dois pacotes por intervalo. No entanto, a média obtida é próxima a três pacotes. O mesmo comportamento ocorre no programa Fonte 1.0, porém com um valor médio menor. Isso faz com que a quantidade de intervalos de On e Off seja menor e que a Fonte 2.0 gere mais tráfego que as demais fontes implementadas.

Neste sentido, na Fonte 2.0 a convergência dos valores não é tão próxima como nos outros modelos. Isso também pode ser justificado pelo fato que, quando um intervalo é gerado e este não é suficiente para o envio de nenhum pacote, gera-se um novo intervalo, desconsiderando o anterior.

3.6 CONCLUSÕES DO CAPÍTULO

O presente capítulo apresentou alguns modelos de tráfego referendados na literatura, especificando os motivos para sua utilização. Com isso, pode-se verificar que o campo de pesquisa

nesta área ainda está em aberto, pois embora o modelo auto-similar seja o mais adequado na atualidade, sua complexidade no desenvolvimento ainda restringe o seu uso. Além disso, boas aproximações ao tráfego auto-similar podem ser realizadas usando o Modelo On/Off.

Em função de sua simplicidade, o modelo On/Off foi adotado neste trabalho, sendo implementados três tipos de fontes, baseadas neste modelo, visando analisar o quanto o tamanho dos pacotes, o tempo de On e de Off, e a taxa de geração de pacotes influencia no desempenho dos sistemas de rede.

Como algumas aplicações, tais como o tráfego telefônico, usam pacotes de tamanho fixo, a Fonte Fixa foi a primeira a ser desenvolvida e analisada. Já a Fonte 1.0 e a Fonte 2.0 geram pacotes com tamanhos médios baseados na distribuição exponencial, sendo que o processo de geração dos pacotes difere entre estas duas fontes, conforme foi descrito anteriormente.

A partir da implementação das fontes, simulações foram realizadas com 100000, 200000 e 300000 pacotes para cada um dos tráfegos, sendo estes referentes a voz, vídeo e dados.

Analisando os resultados das simulações, constatou-se que com 100000 pacotes os programas já estão convergindo, conforme apresentado nas Tabelas 3.2, 3.3 e 3.4. Afinal, a diferença nos valores com a geração de 100000, 200000 e 300000 pacotes é praticamente inexistente.

No entanto, observa-se que a mudança na forma como os pacotes são gerados causa diferença em relação a alguns parâmetros. Na simulação com 100000 pacotes de voz usando o programa Fonte On/Off Fixa, o tempo de chegada do último pacote foi 538.8950 s. Usando o programa Fonte On/Off Variável 1.0, em uma simulação também com 100000 pacotes de voz, sendo estes gerados com quantidade média de dois pacotes por intervalo, o tempo de chegada do último pacote foi 522.6723 s. Já no programa Fonte On/Off Variável 2.0, com as mesmas configurações, o tempo de chegada do último pacote foi 629.6921 s.

Portanto, provavelmente haverá maior ociosidade no servidor usando a fonte On/Off Variável 2.0. No entanto, os outros modelos de fontes propostos podem gerar maior perda de pacotes. Assim, dependendo do tipo de serviço requerido, há um formato de pacote mais

apropriado. Por exemplo, aplicações sensíveis a atrasos, mas tolerantes à perda, como é o caso de aplicações de vídeo, não devem ser realizadas usando uma fonte parecida com a fonte On/Off Variável 2.0.

Finalmente, já se pode inferir que as fontes exercem muita influência sobre os processadores e nós da rede, especialmente em termos do tamanho do pacote e da taxa de geração. Assim, dependendo da QoS desejada, um determinado tipo de fonte deve ser usado para a análise dos escalonadores.

No próximo capítulo será apresentada a implementação dos escalonadores de tráfego *First In First Out* (FIFO), *Deficit Round Robin* (DRR) e o DRR com banda efetiva. A partir de vários cenários, um nó da rede convergente será analisado, para verificar o comportamento e impacto dos tráfegos multimídia. Nesta análise serão considerados o tempo de serviço, o tempo de fila, o tempo de sistema e a taxa de descarte de pacotes.

4 ANÁLISE DE UM NÓ DA REDE CONVERGENTE COM ESCALONADORES FIFO, DRR E DE BANDA EFETIVA

Através da revisão bibliográfica realizada no Capítulo 2, constatou-se que dentre os diversos algoritmos de escalonamento existentes, os escalonadores *First In First Out* (FIFO) e o *Déficit Round Robin* (DRR) são boas soluções. O DRR apresenta enfileiramento justo e é de baixa complexidade se comparado a outros algoritmos mais comumente utilizados em redes convergentes, como o *Weighted Fair Queueing* (WFQ) [7]. O FIFO foi escolhido devido a sua simplicidade. Além disso, “o algoritmo FIFO é amplamente utilizado nos roteadores da Internet em função da sua simplicidade” [86]. O algoritmo FIFO não só é útil para a partilha de atrasos entre as conexões, reduzindo o retardo para uma conexão particular durante seus períodos de rajada, como também é fácil de implementar e é eficiente [7].

Um simulador para cada um dos dois escalonadores foi desenvolvido, visando identificar o impacto que as novas aplicações geram no nó da rede convergente. O modelo do tráfego usado é baseado no tipo de fonte On/Off. Neste tipo de tráfego, há dois períodos: On e Off. Este modelo foi apresentado no Capítulo 3, sendo justificado o motivo da escolha. As distribuições dos comprimentos do período e dos intervalos entre períodos podem ser escolhidas para se adaptarem a cada tipo de tráfego. Embora este tipo de fonte não seja totalmente apropriado ao tráfego real, escolhendo-se parâmetros convenientes, pode se tornar um bom modelo para representar os diferentes tipos de tráfego. Os impactos principais que são investigados referem-se ao tempo de espera, ao tempo de serviço e ao tempo de sistema em um nó de rede, bem como à perda de pacotes.

O capítulo culmina com a apresentação de uma proposta inédita que consiste em uma nova forma de escalonamento de pacotes, que usa o conceito de banda efetiva para a distribuição de quotas para os diferentes fluxos no escalonador DRR. Através da proposta apresentada é possível conter a perda de pacotes, parâmetro importante para satisfazer a QoS do tráfego de dados. Além disso, o novo escalonador também apresenta valores consideráveis no que se refere ao atraso de sistema. No entanto, o atraso ainda precisa ser melhor investigado de modo a satisfazer as exigências dos tráfegos de voz e vídeo.

4.1 ANÁLISE COM O ESCALONADOR FIFO

Para avaliar os impactos provocados no escalonador FIFO devido às alterações nos parâmetros e no processo de funcionamento das fontes, foram realizadas simulações com todas as fontes propostas, usando os parâmetros apresentados na Tabela 3.1 do Capítulo 3.

Para tanto, uma plataforma de simulação foi desenvolvida na linguagem C++ Builder, implementando todos os tipos de fontes apresentados no Capítulo 3, o escalonador FIFO e um buffer que representa o nó da rede (Figura 4.1). Este simulador, descrito no Capítulo 2, foi desenvolvido conforme o fluxograma mostrado na Figura 2.2.

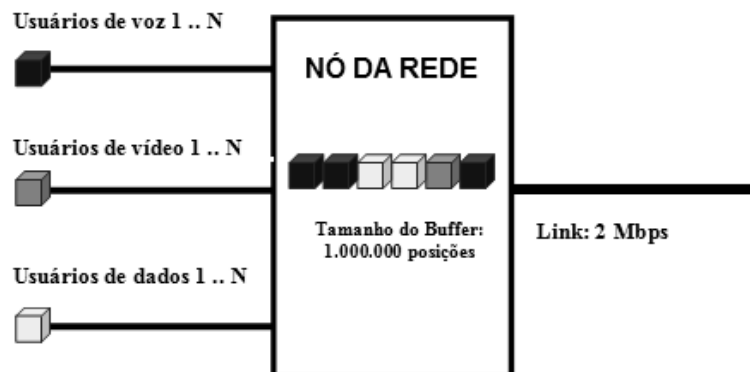


Figura 4.1: Configuração da rede usada nas simulações com o escalonador FIFO

Através da Figura 4.1 pode-se visualizar a configuração do nó de rede analisado. Na entrada do nó de rede, os usuários de voz, vídeo e dados são gerados e um link de 2 Mbps é utilizado na saída. Um buffer muito grande é considerado para simular um buffer infinito.

Este esquema foi alimentado com as fontes descritas no capítulo 3, sendo que para os tráfegos de voz e vídeo apenas a Fonte Fixa foi usada. Para dados, usaram-se as demais fontes relatadas. Vale salientar que, para todas as fontes geradas, foram considerados os parâmetros apresentados na Tabela 3.1, conforme o tipo de tráfego que estava sendo analisado. Além disso, para cada usuário no sistema, independente do tipo de tráfego, foram gerados 50.000 pacotes.

Para estudar o comportamento dos tipos de tráfegos, vários cenários foram analisados, aumentando gradativamente o número de usuários no sistema. Isso permitiu verificar a saturação do enlace físico, bem como o tempo médio de serviço ($E\{X\}$), o tempo médio de fila no buffer ($E\{W\}$), o tempo médio de sistema ($E\{T\}$), equivalente a $E\{W\} + E\{X\}$ e, a Perda de Pacotes (PP).

Desta forma, os cenários considerados são apresentados na Tabela 4.1.

Tabela 4.1: Cenários considerados

Cenário	Tráfego	Tipos de Fontes	Quantidade de Usuário Inicial	Quantidade de Usuário Final
1	Dados	Fonte 1.0	5	15
		Fonte 2.0	5	15
2	Dados	Fonte 1.0	4	4
	Voz	Fonte 2.0	4	4
		Fonte Fixa	1	16
3	Dados	Fonte 1.0	4	4
	Vídeo	Fonte 2.0	4	4
		Fonte Fixa	1	6
4	Dados	Fonte 1.0	4	4
	Voz	Fonte 2.0	4	4
		Fonte Fixa	5	5
	Vídeo	Fonte Fixa	1	3
5	Dados	Fonte 1.0	4	4
	Voz	Fonte 2.0	4	4
		Fonte Fixa	1	13
	Vídeo	Fonte Fixa	1	1
6	Dados	Fonte 1.0	4	4
	Voz	Fonte 2.0	4	4
		Fonte Fixa	1	5
	Vídeo	Fonte Fixa	2	2

Através da Tabela 4.1 pode-se constatar que seis cenários foram analisados, permitindo confrontar o comportamento do desempenho da rede, que opera basicamente com o tráfego de

dados (Cenário 1), e a performance com tráfego multimídia, caso os recursos utilizados sejam os mesmos.

Também se observa, através da Tabela 4.1, que a partir do segundo cenário em todas as simulações há quatro usuários de dados no sistema, variando o(s) outro(s) tráfego(s). Por exemplo, no Cenário 5 todas as simulações foram realizadas com quatro usuários de dados, um usuário de vídeo e aumentando gradativamente a quantidade de usuários de voz.

Conforme mencionado, as simulações foram realizadas considerando um enlace físico de 2 Mbps, descartando-se o regime transitório. Para isso adotou-se o procedimento de simular, por um espaço de tempo antes de iniciar a coleta estatística dos dados para fins de relatório, desconsiderando os primeiros 1.000 pacotes.

4.1.1 Resultados dos Cenários Simulados com o Escalonador FIFO

Para analisar o comportamento do tráfego multimídia em um nó da rede, estudou-se inicialmente a rede somente com o tráfego de dados. Neste cenário é possível verificar os pontos de saturação (valores de usuários em que o atraso de sistema tende a ser muito grande) para as condições específicas de rede, considerando somente o tráfego de dados, que representa a maior parte do tráfego, atualmente. Nos cenários seguintes, estudou-se o comportamento da rede, considerando o crescimento dos tráfegos de vídeo e/ou de voz, em relação ao tráfego de dados.

Na Figura 4.2 são apresentados os resultados das simulações para o Cenário 1, utilizando as Fontes 1.0 e 2.0.

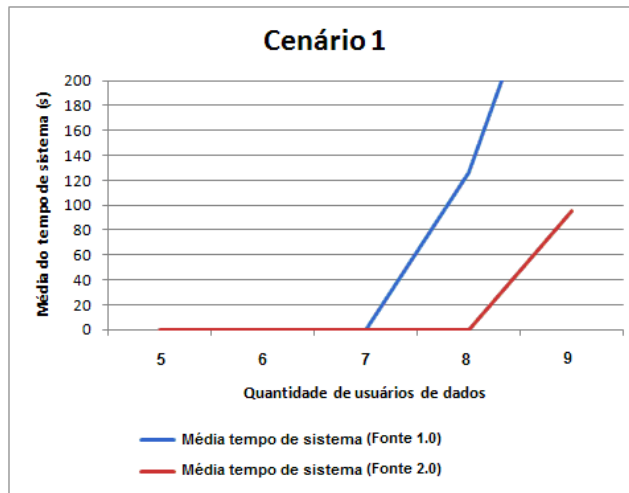


Figura 4.2: Simulação com aumento de usuários de dados

Conforme pode ser observado pela Figura 4.2, o tempo de atraso de sistema é menor na simulação com a Fonte 2.0. O mesmo ocorre no que se refere ao tempo de fila e de serviço. Isso pode ser explicado pela maneira como os pacotes são gerados em cada fonte. No caso da Fonte 1.0, todos os pacotes gerados em um T_{on} são enviados, enquanto que no caso da Fonte 2.0, às vezes, o pacote gerado não cabe naquele T_{on} e é enviado no T_{on} seguinte, gerando, dessa forma, um tráfego menor. Deste modo, com a Fonte 2.0 o servidor fica mais tempo ocioso, pois os pacotes sejam em tempos mais espaçados, como já foi relatado no capítulo anterior. Os valores de tempos de sistema antes da saturação, para os dois casos, são valores pequenos e praticamente iguais, variando de 2 a 8 ms.

Outro fator importante deste cenário é que os valores dos tempos de sistema têm grande tendência de aumento somente a partir das simulações com sete e oito usuários para as Fontes 1.0 e 2.0, respectivamente. Estes números de usuários são valores onde o sistema entra em saturação, significando que o atraso tende a ser muito grande e pode haver um engarrafamento total. Uma operadora de rede deve evitar, a todo custo, esses pontos de saturação.

No tocante ao tempo de serviço, com nove usuários, este parâmetro não se submete a nenhum aumento significativo, como esperado, porque o tempo de serviço depende somente da média do tamanho dos pacotes e da capacidade do enlace físico.

Além disso, é importante mencionar que independente do modelo de fonte usado, nenhuma perda de pacotes ocorreu.

Pode-se observar pela Figura 4.2 que, com cinco usuários, o sistema fica bastante estável, sem nenhuma possibilidade de engarrafamento. Dessa forma, para as análises dos cenários seguintes, um grupo de quatro usuários de dados foi mantido constante, e a quantidade de outros usuários foi variada para possibilitar a análise das influências desses usuários no sistema.

No segundo cenário (Cenário 2) analisado, todas as simulações foram realizadas com quatro usuários de dados, sendo estes gerados com a Fonte 1.0 e Fonte 2.0, e os usuários de voz, gerados com a Fonte Fixa, foram aumentados gradativamente. Assim, a quantidade de pacotes de dados gerados foi de 200.000 pacotes em todas as simulações. No entanto, o número de pacotes de voz gerados iniciou com 50.000 pacotes na primeira simulação, com apenas um usuário de voz, e na última simulação, 800.000 pacotes de voz foram gerados.

Os resultados dessas simulações são apresentados na Figura 4.3.

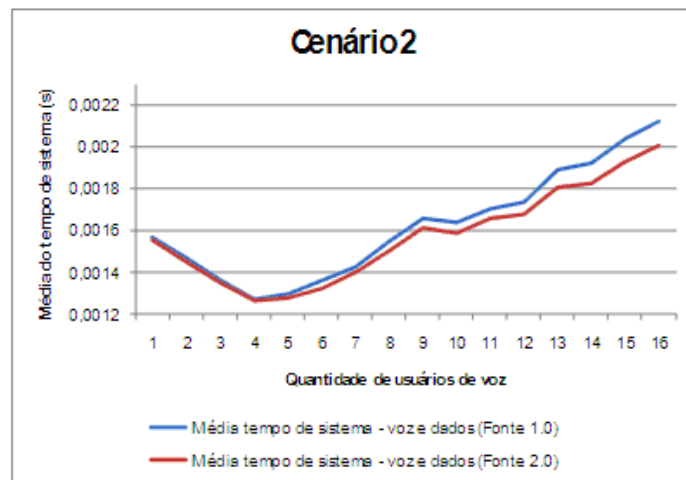


Figura 4.3: Simulação com aumento de usuários de voz

Através da Figura 4.3 pode-se observar que o aumento de usuários de voz provoca um crescimento lento do tempo de sistema, devido ao baixo tráfego de cada usuário de voz. Isso permite concluir que o sistema suporta um bom volume de tráfego de voz, sem atingir a região de

saturação. A região de saturação pode ser observada nas figuras no instante em que o incremento de um único usuário no sistema causa uma variação muito grande no tempo de sistema.

Se comparado ao Cenário 1, percebe-se que os tempos de sistemas são menores do que no Cenário 2. Isso ocorre em função que no Cenário 1 só há tráfego de dados e estes têm pacotes maiores. Já no Cenário 2, como os pacotes de voz são menores e há poucos pacotes de dados, o tempo de sistema é menor.

Outro ponto observado é que o aumento de usuários de voz reduz o tempo de serviço em função do tamanho dos pacotes. Isso ocorre até quatro usuários de voz no sistema. Após, o tempo de serviço praticamente não sofre alteração, devido ao pequeno tamanho dos pacotes de voz.

No terceiro cenário (Cenário 3), foram feitas simulações mantendo-se a quantidade de usuários de dados fixa em quatro usuários, sendo estes gerados com a Fonte 1.0 e 2.0, e aumentando-se a quantidade de usuários de vídeo, começando com um usuário e finalizando com seis usuários. Os pacotes de vídeo foram gerados com a Fonte Fixa.

Os resultados dessas simulações estão apresentados na Figura 4.4.

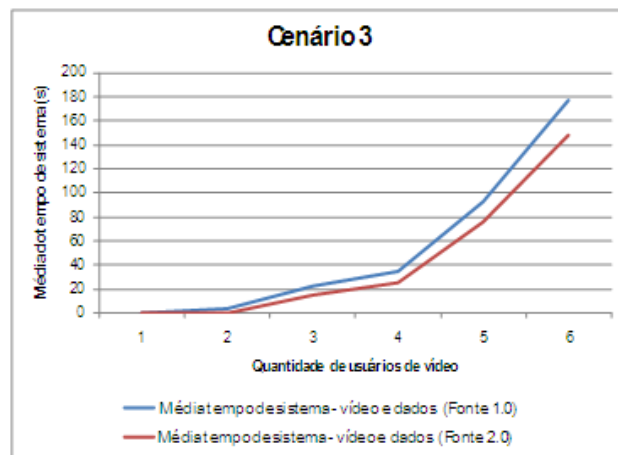


Figura 4.4: Simulação com aumento de usuários de vídeo (Cenário 3)

Analisando os resultados apresentados na Figura 4.4, observa-se que o aumento de usuários de vídeo provoca o aumento gradual do tempo de sistema, sendo mais acentuado a partir de quatro usuários. Entretanto, comparando a Figura 4.4 com a Figura 4.2, o aumento do tempo de sistema é significativo a partir de três usuários de vídeo. Isso é devido à natureza mais intensa do tráfego de

vídeo. Porém, o aumento não é tão drástico como no caso de somente dados, que entra em saturação repentinamente com oito usuários de dados. Supõe-se que esse fenômeno ocorre devido aos pacotes de vídeo terem comprimento constante e não aleatório, como no caso dos pacotes de dados. O aumento do tempo de sistema é mais suave, mas considerável a partir de cinco usuários de vídeo.

Com relação ao tempo de serviço, não há nenhuma variação significativa porque o tempo de serviço de vídeo e o tempo de serviço de dados são, em média, o mesmo valor, permanecendo no valor médio.

No quarto cenário (Cenário 4) foram feitas simulações com pacotes de voz, vídeo e dados. Os dois primeiros tipos de pacotes foram gerados usando a Fonte Fixa, enquanto o último foi gerado com as Fontes 1.0 e 2.0. Além disso, todas as simulações tiveram uma quantidade fixa de cinco usuários de voz e quatro usuários de dados, enquanto que a quantidade de usuários de vídeo sofreu alterações. Assim, foi gerado um total de 250.000 pacotes de voz e 200.000 pacotes de dados por simulação. No entanto, o número de usuários de vídeo foi aumentado gradualmente, sendo 50.000 pacotes para cada usuário. Assim, na simulação com um usuário de vídeo, 50.000 pacotes de vídeo foram gerados, enquanto com dois e três usuários de vídeo 100.000 e 150.000 pacotes de vídeo, respectivamente, foram gerados.

A Figura 4.5 mostra os resultados obtidos neste cenário.

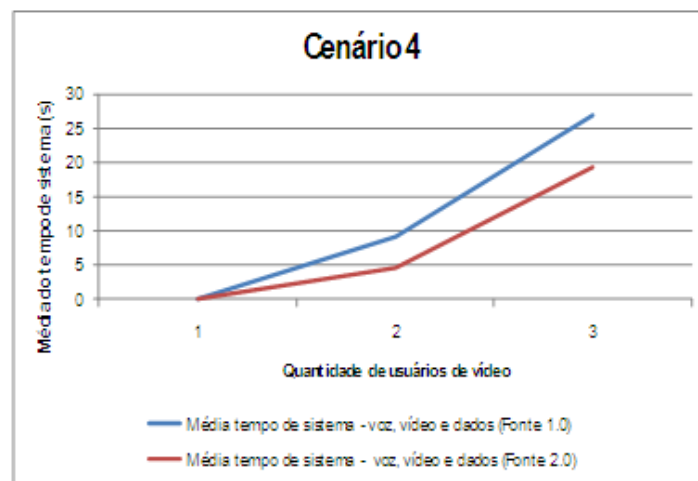


Figura 4.5: Simulação com aumento de usuários de vídeo (Cenário 4)

Analisando os resultados apresentados na Figura 4.5, observa-se que tanto nas simulações em que os pacotes de dados são gerados com a Fonte 1.0, quanto nas simulações com a Fonte 2.0, o aumento de usuários de vídeo provoca o aumento do tempo de sistema. Entretanto, esse aumento é mais intenso nas simulações com a primeira fonte citada.

Nota-se o mesmo fenômeno observado na Figura 4.4 que é o rápido alcance do ponto de saturação (aumento acentuado no tempo de sistema) com o aumento de usuários de vídeo, evidenciando a grande influência que esses usuários podem ocasionar no sistema.

Consoante ao comportamento do tempo de serviço, já inicia com valor alto, em função da presença dos tráfegos de dados, voz e vídeo. Aumentando-se o número de usuários de vídeo, o tempo de serviço aumenta igualmente. É importante mencionar que embora o tempo de serviço não esteja sendo apresentado nos gráficos, o mesmo foi analisado, assim como o tempo de fila.

No quinto cenário (Cenário 5) os pacotes de vídeo e voz foram gerados de acordo com a Fonte Fixa, enquanto que os pacotes de dados foram gerados pelas Fontes 1.0 e 2.0. Porém, neste cenário, variou-se a quantidade de usuários de voz, enquanto que a quantidade de usuários de vídeo foi de um usuário e a quantidade de dados foi de quatro usuários em todas as simulações.

Os resultados dessas simulações são apresentados na Figura 4.6.

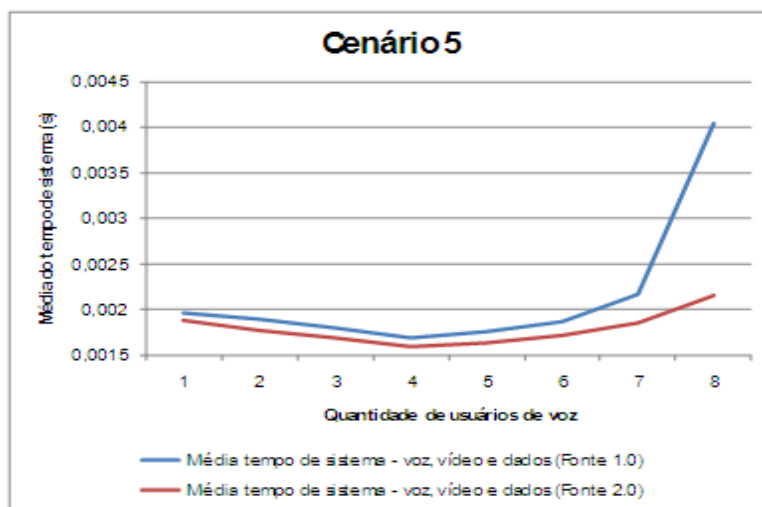


Figura 4.6: Simulação com aumento de usuários de voz (Cenário 5)

Conforme pode ser observado na Figura 4.6, neste cenário o crescimento do tempo de sistema é bastante lento como ocorreu no Cenário 2, cujos resultados são mostrados na Figura 4.3. Como os tamanhos dos pacotes de voz são menores que os pacotes de vídeo e dados, o impacto de seu crescimento é menor, pois significa uma intensidade menor de tráfego. Pode-se concluir novamente que o sistema admite um bom volume de tráfego de voz sem atingir o ponto de saturação.

No que tange ao tempo de serviço, neste cenário ele é quase constante, começando em um valor mais elevado, em função dos tráfegos de vídeo e dados, e reduzindo ligeiramente com o aumento do tráfego de voz devido ao tamanho dos pacotes de voz ser menor. Embora a quantidade de pacotes no sistema seja maior, como os pacotes de voz são pequenos, o aumento de um usuário de voz não ocasiona mudanças significativas no sistema.

No sexto cenário (Cenário 6), as simulações realizadas foram semelhantes às simulações do Cenário 5. O que difere é a quantidade de usuários de vídeo que, neste cenário foram utilizados uma quantidade fixa de dois usuários.

Os resultados dessas simulações são exibidos na Figura 4.7.

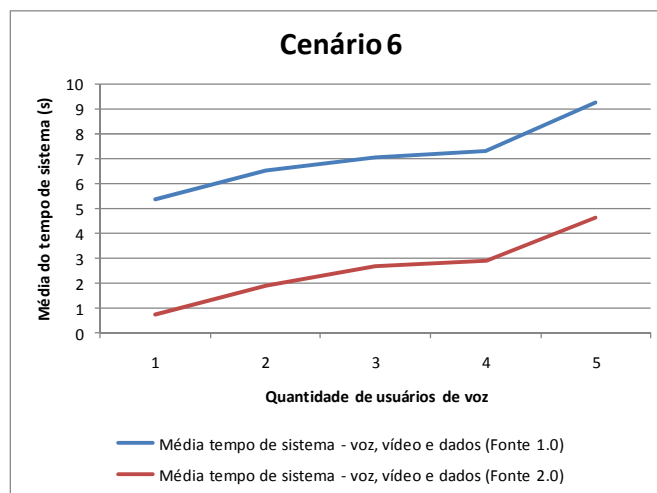


Figura 4.7: Simulação com aumento de usuários de voz (Cenário 6)

Através da Figura 4.7 percebe-se que apesar das semelhanças nas simulações entre o quinto e o sexto cenário, estes geraram resultados com grandes diferenças em alguns aspectos. O principal

contraste entre os dois cenários é na diferença entre os tempos de sistema, que neste caso já são valores consideráveis. Esse comportamento ocorreu porque ao aumentar a quantidade de usuários de vídeo para dois, o conseqüente aumento da quantidade de pacotes de vídeo provocou grande aumento nos tempos de fila das simulações, já que os pacotes de vídeo são bem maiores do que os pacotes de voz. Desta forma, os tempos de sistema acompanharam a mesma tendência, pois seu valor depende do tempo de fila.

Entretanto, um ponto importante a ser observado neste cenário é que a influência do tráfego de voz torna-se significativa com o aumento de usuários de voz. Como o sistema já está operando perto da região de saturação, pequenos incrementos de tráfego de voz tendem a ter bastante impacto no tempo de sistema. Assim, este cenário mostra a importância de controle de acesso de novos usuários. Como na rede atual não há nenhum controle de acesso, o sistema poderá sofrer engarrafamento. Este cenário mostra, também, a importância de se ter algum mecanismo para garantir a qualidade de serviços dos usuários. O tráfego de voz deve ser tratado quase em tempo real para não ocasionar problemas de comunicação. No caso em que houver atrasos significativos na transmissão de pacotes de voz, a comunicação é dificultada, pois, a pergunta ou a resposta demora a chegar, ocasionando dificuldades na conversação. Este Cenário 6 mostra que os atrasos no sistema são iguais tanto para dados, vídeo e voz. Mas, esses atrasos já são proibitivos para usuários de voz, que suportam atrasos máximos em torno de 200 ms, para uma conversação natural.

Em relação à perda de pacotes, independente do cenário ou modelo de fonte usado como entrada de dados no simulador para escalonamento FIFO, nenhuma perda de pacotes ocorreu. Isso se deve ao fato de que o tamanho do buffer definido em todas as simulações apresentadas foi de 1.000.000 de posições. Uma maneira de controlar o tempo de sistema com a redução desse tempo, seria diminuir o tamanho do buffer, gerando perda de pacotes. Entretanto, embora essa técnica de controle seja aceitável para transmissão de dados que utiliza retransmissão (por ex., *Transmission Control Protocol* (TCP)), não é razoável para voz que utiliza o *User Datagram Protocol* (UDP), pois perdas elevadas ocasionam frases entrecortadas, difíceis de compreender.

Através das simulações e cenários analisados, pode-se concluir, o que já era esperado, que o aumento do tráfego de vídeo tem grande impacto nos sistemas de redes, pois em todas as simulações esse crescimento provocou grande aumento no tempo de sistema.

O crescimento do tráfego de voz provoca impactos menores para o sistema operando em condições estáveis. Entretanto, se o sistema está operando na região perto do ponto de saturação, o seu incremento poderá levar ao ponto crítico, que é o engarrafamento total da rede. Além disso, comprovou-se que esse tipo de tráfego deve ser tratado de maneira especial, por exemplo, dando prioridade no seu tratamento, para que não haja atrasos consideráveis que dificultam a comunicação entre os seus interlocutores. Ou seja, há comprovação da necessidade de prover mecanismos que garantam a qualidade de serviço na rede atual.

4.1.2 Conclusões Referentes ao Escalonador FIFO

Neste tópico, o impacto das fontes de tráfego multimídia em um nó de rede usando o escalonador FIFO foi estudado. A intenção da pesquisa foi analisar como os tráfegos de voz e de vídeo podem impactar a atual rede que se baseia no escalonador FIFO e não tem nenhum mecanismo de controle de acesso e de qualidade do serviço. Recentemente, esses dois tipos de tráfegos têm crescido com grande intensidade.

Para atingir os objetivos propostos, uma plataforma de simulação que consiste em diferentes tipos de fontes, um buffer e um escalonador FIFO usando um link de 2 Mbps foi desenvolvida, e seis cenários diferentes, com várias combinações de voz, vídeo e tráfego de dados foram analisados. Através das simulações foi possível verificar o comportamento dos diferentes tráfegos em relação ao tempo de sistema, supondo não ocorrer descarte de pacotes.

Analisando-se os cenários propostos, concluiu-se que o tráfego de vídeo ocasionou um crescimento rápido no tempo de sistema. Essa conclusão ficou mais evidente nos Cenários 3 e 4, nos quais o incremento de um único usuário de vídeo causou um aumento superior a 5 s no tempo de

sistema. Além disso, constatou-se que três usuários de vídeo no sistema (Cenário 4), podem levar o sistema ao ponto de saturação.

Confrontando o Cenário 6 com os Cenários 2 e 5, os quais analisaram o comportamento do aumento de usuários de voz, concluiu-se que se o sistema estiver na região estável, não há um impacto significativo nos tempos de sistema, possibilitando a inclusão de um grande número de usuários deste tipo de tráfego. No entanto, se o sistema estiver perto do seu ponto de saturação, o que ocorre no sexto cenário, um usuário a mais já causa um aumento expressivo no tempo de sistema.

Uma conclusão importante que os estudos dos cenários propostos proporcionaram é a necessidade de algum tipo de controle de admissão para não prejudicar o desempenho dos tráfegos em andamento, bem como não congestionar a rede. Além disso, deve ser providenciado algum tipo de escalonador que garanta a qualidade de serviço. Esse escalonador pode ser do tipo *Deficit Round Robin* (DRR) [111], ou *Weighted Fair Queuing* (WFQ) [33], entre outros. Estes escalonadores são destinados a garantir a equidade e/ou a *Quality of Service* (QoS) de cada tipo de tráfego.

Na próxima seção é analisado o impacto do escalonador DRR em um nó de rede convergente.

4.2 ANÁLISE COM O ESCALONADOR DRR

Como a proposta é verificar a equidade na distribuição dos recursos, um simulador para o escalonamento DRR foi implementado usando a linguagem C++ Builder, sendo que o mesmo foi alimentado com as fontes propostas e apresentadas no Capítulo 3.

O escalonador DRR foi escolhido por ser um algoritmo bem difundido e já consolidado na literatura, por sua simplicidade computacional, por permitir distribuir cotas diferenciadas para os diversos tipos de tráfego, atendendo desta forma a QoS requerida pelos usuários e aplicações. Além

disso, há facilidade de modificar este algoritmo de escalonamento, de modo a obter bons resultados no que se refere ao atraso e perda de pacotes, como pode ser comprovado em [26] e [102].

O intuito do simulador desenvolvido é verificar a saturação de um nó de rede de 2 Mbps quando há apenas tráfego de dados, confrontando os resultados com cenários onde há um aumento de tráfego multimídia.

A configuração do nó usado na plataforma de simulação é mostrada na Figura 4.1.

Conforme já relatado e observado através da Figura 4.1, o nó de rede recebe como entrada os tráfegos gerados pelos usuários de voz, vídeo e dados, e os pacotes recebidos são armazenados em um buffer de 1.000.000 posições (praticamente um buffer de tamanho infinito) e são processados por um link de saída de 2 Mbps. Para o caso do escalonador DRR, é providenciado um buffer para cada um dos tráfegos.

O objetivo é, inicialmente, alimentar o nó somente com o tráfego de dados e estudar o seu comportamento em um enlace de 2 Mbps. É um cenário que simula o comportamento da rede para a situação ocorrida até recentemente que operava somente com o tráfego de dados. Nos outros cenários, introduzem-se os tráfegos de voz e vídeo, e se estuda o comportamento do nó. Isso permitirá analisar o impacto que as aplicações de voz e vídeo causam no sistema.

Neste sentido, os tráfegos de voz e vídeo foram gerados usando apenas a Fonte Fixa. Para dados, usou-se as Fontes 1.0 e 2.0. Vale salientar que em todas as simulações, os parâmetros utilizados para gerar as fontes foram os apresentados na Tabela 3.1, presente no terceiro capítulo, conforme o tipo de tráfego em questão.

Através do simulador pode-se analisar a saturação do enlace físico, o tempo médio de serviço que corresponde ao tamanho do pacote dividido pela capacidade do canal, o tempo médio de espera no buffer, tempo médio no sistema, que corresponde ao somatório do tempo médio de fila e tempo médio de serviço, bem como a perda de pacotes. No entanto, é importante salientar que o simulador foi desenvolvido com um buffer de 1.000.000 de posições para cada um dos tráfegos, sendo uma analogia a um buffer infinito.

A definição dos cenários é mostrada na Tabela 4.2. Como se pode perceber, a quantidade de usuários inicial é baixa, sendo aumentada gradualmente até o número de usuários definido em quantidade final de usuários. Para cada usuário, 50 mil pacotes são gerados. Assim, se existem quatro usuários no sistema, 200 mil pacotes serão processados pelo escalonador. Desta forma, no sexto cenário, o tráfego inicial será de 350 mil pacotes, pois há sete usuários no sistema: quatro de dados, um de voz e dois de vídeo.

As simulações com o escalonador DRR foram geradas inicialmente considerando as seguintes quotas: 4096 bits para o tráfego de dados e de vídeo e 1360 bits para o tráfego de voz. Estas simulações são identificadas como A na Tabela 4.2. As simulações subsequentes foram realizadas com quotas de 2720 bits para o tráfego de voz, 6144 bits para o tráfego de vídeo e 4096 bits para o tráfego de dados. Estas simulações são representadas por B. Os valores para as quotas nas simulações A foram escolhidos porque correspondem ao tamanho médio dos pacotes para estes tipos de tráfego. Assim, há grande probabilidade de que em um ciclo possa ser enviado pelo menos um pacote. No entanto, como a intenção é priorizar os tráfegos de voz e de vídeo, a configuração B foi, também, analisada.

Tabela 4.2: Cenários considerados para análise com o escalonador DRR

Cenários	Tráfegos	Tipos de fontes	DRR e FIFO	DRR		FIFO
			Quantidade Inicial de Usuários	Quantidade Final de Usuários - A	Quantidade Final de Usuários - B	Quantidade Final de Usuários
1	Dados	Fonte 1.0	1	9	9	15
		Fonte 2.0	1	9	9	15
2	Dados	Fonte 1.0	4	4	4	4
		Fonte 2.0	4	4	4	4
	Voz	Fonte Fixa	1	15	15	16
3	Dados	Fonte 1.0	4	4	4	4
		Fonte 2.0	4	4	4	4
	Vídeo	Fonte Fixa	1	5	8	6
4	Dados	Fonte 1.0	4	4	4	4
		Fonte 2.0	4	4	4	4
	Voz	Fonte Fixa	5	5	5	5
	Vídeo	Fonte Fixa	1	3	7	3
5	Dados	Fonte 1.0	4	4	4	4
		Fonte 2.0	4	4	4	4
	Voz	Fonte Fixa	1	9	15	13
	Vídeo	Fonte Fixa	1	1	1	1
6	Dados	Fonte 1.0	4	4	4	4
		Fonte 2.0	4	4	4	4
	Voz	Fonte Fixa	1	3	15	5
	Vídeo	Fonte Fixa	2	2	2	2

Como se pode constatar através da Tabela 4.2, o primeiro cenário se restringe ao tráfego de dados. Pode-se antecipar que, com nove usuários, que correspondem a 450 mil pacotes no sistema, haverá uma saturação do link de 2 Mbps, tanto para o escalonador DRR quanto para o escalonador FIFO. Nos outros cenários, o tráfego de dados é fixado em quatro usuários, o que corresponde a 200 mil pacotes deste tipo de tráfego, e foi variado o número de usuários de voz e vídeo. Adotou-se o valor de quatro usuários de dados, pois com este número o link está estável, permitindo aceitar conexões de voz e vídeo. O objetivo do segundo até o sexto cenário é verificar o impacto que o uso de aplicações multimídia gera no link, se mantidos os recursos utilizados.

Comparando os escalonadores DRR e FIFO, através da análise da Tabela 4.2, já é possível antecipar que o escalonador DRR admite um maior número de usuários no sistema, exceto no Cenário 1, onde há apenas o tráfego de dados. No Cenário 1, o comportamento de ambos os escalonadores é similar. Supõe-se que a pequena diferença favorável ao FIFO seja devida às variações estatísticas dos dados obtidos na simulação. Pode-se concluir que para a Internet somente com o tráfego de dados, o FIFO era bastante adequado.

Analisando o DRR, na Tabela 4.2, pode ser visto que, com quotas maiores, de acordo com a configuração B, existe a possibilidade de uma maior quantidade de usuários no sistema. Isso mostra a influência da quota no desempenho do escalonador DRR. Assim, apenas os resultados obtidos com a configuração B são apresentados graficamente na análise dos resultados. Outrossim, os cenários serão analisados considerando os seguintes parâmetros: tempo médio de serviço, tempo médio de espera na fila, tempo médio do sistema, bem como a perda de pacotes.

Para o cálculo do intervalo de confiança, algumas simulações foram executadas três vezes. Desta forma, a confiabilidade nos resultados é assegurada em 68,3% (um desvio padrão em torno do valor médio), com um coeficiente t-Student de 1,32. Além disso, foi descartado o período transitório. Para isso, os primeiros 1.000 pacotes foram desconsiderados para fins estatísticos.

4.2.1 Resultados Comparando os Escalonadores FIFO e DRR

Para analisar o comportamento do tráfego, foram feitas simulações considerando os cenários apresentados na Tabela 4.2. Desta forma, no primeiro cenário foi analisado o comportamento da rede somente com o tráfego de dados, aumentando a quantidade de usuários deste tipo de tráfego. Já, nos outros cenários, analisou-se o comportamento da rede mediante o crescimento dos tráfegos de vídeo e de voz, em relação ao tráfego de dados.

A Figura 4.8 apresenta os resultados referentes à simulação com o primeiro cenário proposto, utilizando-se para a geração do tráfego de dados a Fonte 1.0 e a Fonte 2.0 no escalonador DRR.

É importante mencionar que neste cenário, como há apenas o tráfego de dados, a quota é integral para este tráfego. Neste sentido, o funcionamento do escalonador DRR é similar ao FIFO, pois não há necessidade de distinção entre os fluxos.

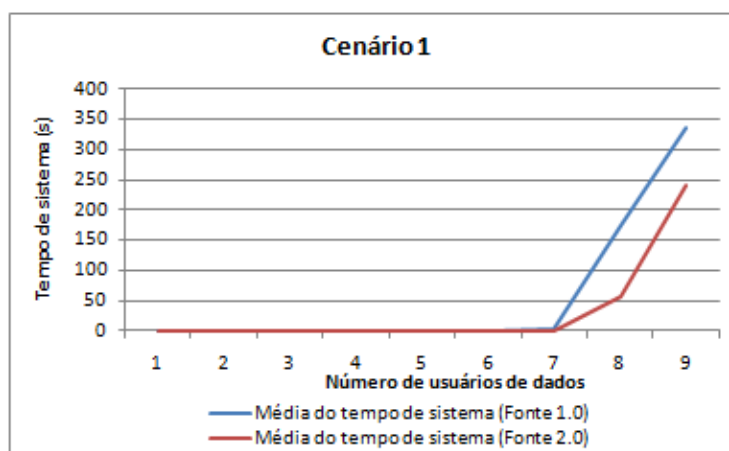


Figura 4.8: Simulação com o aumento de usuários de dados no escalonador DRR

Através da Figura 4.8 pode-se observar que o aumento de usuários de dados só provoca grande influência no valor do tempo de sistema a partir do sétimo usuário, pois com sete usuários os tempos de fila e de sistema não haviam passado de 1 s. No entanto, o aumento do valor desses parâmetros passa a ser bem considerável, pois o tempo de fila, nas simulações com a Fonte 1.0,

passou de 0,90 s na simulação com sete usuários para 174,38 s na simulação com oito usuários. Depois desse valor, o aumento passa a ser cada vez maior entre um usuário e outro.

Também é importante observar que as simulações que utilizam a Fonte 2.0 para a geração dos pacotes de dados possuem menores valores em relação ao tempo de sistema. Por exemplo, com oito usuários, nas simulações com a Fonte 1.0, o valor do tempo de fila foi de 174,38 s, enquanto que nas simulações com a Fonte 2.0, esse valor é de 55,89 s. Essa grande diferença pode ser explicada porque na Fonte 1.0 mais pacotes são enviados em um mesmo ciclo do simulador DRR. Esse fator provoca um tráfego maior e conseqüentemente maior tempo de fila e de sistema. Já, nas simulações com a segunda fonte, menos pacotes são enviados em um mesmo ciclo ou às vezes é necessário mais de um ciclo para enviar um pacote, devido à limitação da quota. Logo, o tráfego é menor e conseqüentemente é gerado menor tempo de fila e de sistema.

Como o intuito deste tópico, também, é realizar um comparativo entre os escalonadores FIFO e DRR, as barras nas figuras representam os tempos de sistema do algoritmo DRR e as linhas, os tempos de sistema do algoritmo FIFO. Para representar os valores, em alguns cenários somente um eixo é utilizado, pois os tempos de fila tanto do FIFO quanto do DRR são próximos. Quando esses tempos divergem, dois eixos são utilizados para apresentar os valores, sendo os da esquerda relativos aos tempos do algoritmo FIFO e o eixo da direita para o algoritmo DRR. Todos os tempos estão apresentados em segundos.

Assim, a Figura 4.9 apresenta os resultados das simulações para o primeiro cenário, em que foi analisado o comportamento da rede diante de aplicações com tráfego de dados, aumentando-se gradativamente a quantidade de usuários de dados e utilizando a Fonte 1.0 para a geração dos pacotes.

Vale salientar que o funcionamento dos escalonadores FIFO e DRR devem ser similares, tendo em vista que só há tráfego de dados e, deste modo, o DRR não faz distinção entre os tráfegos.

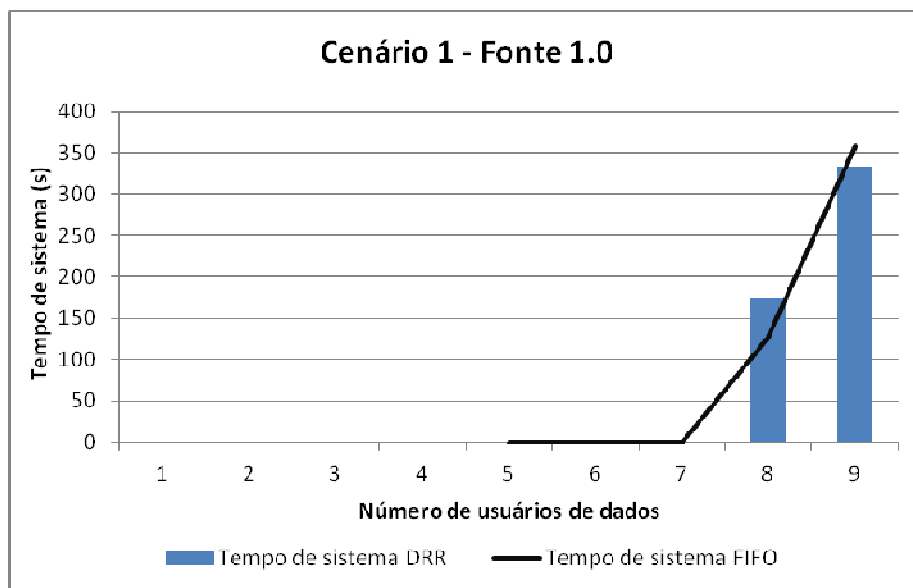


Figura 4.9: Simulações com o aumento de usuários de dados e Fonte 1.0

Através da Figura 4.9 pode-se observar que o aumento de usuários de dados só influencia o tempo de sistema, de forma considerável, a partir do sétimo usuário. Isso tanto no algoritmo DRR quanto no FIFO.

Até sete usuários, o tempo de sistema não havia passado de 1 s, sendo que a partir das simulações com sete usuários houve um rápido e grande aumento nos tempos de fila e de sistema. Por exemplo, no intervalo entre as simulações com sete e oito usuários, com o escalonador DRR, houve um aumento do tempo de sistema de 173,48 s e um aumento do tempo de fila de 173,47 s. Já no escalonador FIFO o aumento no tempo de fila e de sistema foi de 126,62 s no intervalo entre as simulações com sete e oito usuários. Observa-se, portanto, que este aumento de um único usuário levou o sistema ao ponto de instabilidade em que o tempo de sistema pode aumentar indefinidamente e conduzir à situação de engarrafamento total. É uma situação que se deve evitar a todo custo para não haver a parada operacional do sistema.

Além disso, pelas diferenças mencionadas, não há uma alteração significativa, neste cenário, entre os escalonadores analisados.

No Cenário 1, também foram feitas simulações com a Fonte 2.0, considerando o aumento, apenas, do tráfego de dados na rede. Essas simulações estão representadas graficamente na Figura 4.10.

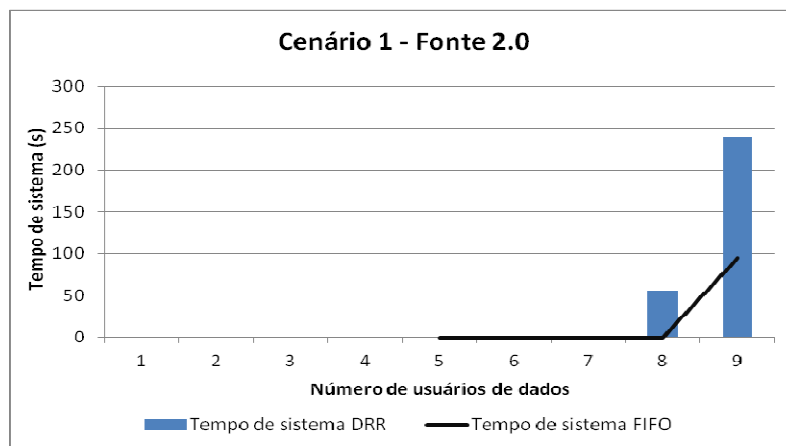


Figura 4.10: Simulações com o aumento de usuários de dados e Fonte 2.0

Observa-se pela Figura 4.10 que o aumento da quantidade de usuários de dados só provoca grande influência no sistema depois das simulações com sete usuários, conforme ocorreram também nas simulações apresentadas na Figura 4.9. Porém esse aumento é inferior ao que ocorreu nas simulações com a Fonte 1.0. Um exemplo disso é que com o escalonador DRR, na simulação com nove usuários de dados e com a Fonte 1.0 o valor do tempo de sistema foi de 334,27 s enquanto que na simulação com a Fonte 2.0 esse valor foi de 240,25 s. Observa-se que o aumento dos usuários de dados provoca maior impacto nas simulações que utilizam a Fonte 1.0 do que nas simulações com a Fonte 2.0, o que já era esperado em função do modo como as referidas fontes são geradas.

Assim, como apenas a fonte geradora foi alterada, os resultados com ambos os escalonadores são similares, conforme pode ser comprovado observando a Figura 4.9 e a Figura 4.10.

Como o objetivo do trabalho é verificar o impacto que as fontes de vídeo e voz causam no sistema, caso sejam mantidos os mesmos recursos, os demais cenários analisados consistem no aumento gradativo dos tráfegos de voz, vídeo, ou ambos, enquanto a quantidade de usuários de dados permanece constante.

Como tanto na Fonte 1.0 como na Fonte 2.0 com quatro usuários de dados o sistema se mantém em perfeito equilíbrio, esse valor foi fixado para o tráfego de dados nos demais cenários.

No segundo cenário analisado foram feitas simulações com o aumento gradativo de usuários de voz, iniciando com um usuário no sistema e finalizando com dezesseis usuários. Porém foi mantida a quantidade de quatro usuários de dados em todas as simulações. O intuito foi verificar o impacto do crescimento de usuários de voz nos tráfegos de redes usando o escalonador DRR. Desta forma, a Figura 4.11 apresenta as simulações com o Cenário 2, utilizando a Fonte 1.0 e Fonte 2.0 para gerar os pacotes de dados, e a Fonte Fixa para a geração do tráfego de voz.

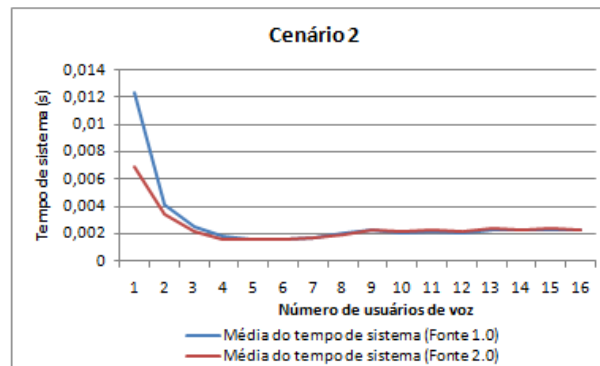


Figura 4.11: Simulação com aumento de usuários de voz e quantidade fixa de usuários de dados no escalonador DRR

Conforme pode ser observado pelo gráfico apresentado na Figura 4.11, o aumento de usuários de voz gera uma queda no tempo de sistema das simulações. No entanto, essa queda ocorre apenas entre as simulações com um até cinco usuários. A partir das simulações com quatro usuários a variação entre os valores do tempo de sistema é baixa. Isso se deve ao tamanho dos pacotes, que a partir de quatro usuários passa a ser menor, pois haverá apenas aumento de pacotes de voz no sistema. Desta forma, pode-se concluir que o sistema torna-se estável, mesmo diante do aumento de usuários de voz.

Analisando o segundo cenário em relação às Fontes 1.0 e 2.0 pode-se observar que o comportamento das simulações é semelhante em relação às duas. No entanto, os valores dos tempos de sistema são menores nas simulações com a Fonte 2.0, até a simulação com nove usuários. Mas a partir desta simulação, o tempo de sistema da Fonte 1.0 passa a ser menor.

Em função do comportamento similar entre as Fontes 1.0 e 2.0, a análise comparativa entre os resultados gerados pelos escalonadores FIFO e DRR será restrita a Fonte 1.0 para a geração do tráfego de dados.

Assim, a Figura 4.12 apresenta as simulações do Cenário 2 comparando os escalonadores FIFO e DRR, considerando que a fonte utilizada para a geração dos pacotes de dados foi a Fonte 1.0, e a Fonte Fixa para o tráfego de voz.

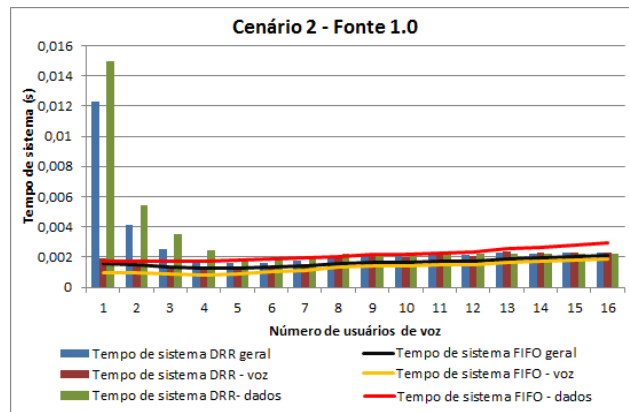


Figura 4.12: Simulações com o aumento de usuários de voz e quantidade fixa de quatro usuários de dados

Conforme pode ser observado pela Figura 4.12, no escalonador DRR, o aumento de usuários de voz provoca uma queda no valor do tempo de sistema entre as simulações com um e cinco usuários. Isso ocorre porque o tamanho dos pacotes de dados é maior e, inicialmente, este tráfego predomina no sistema. Contudo, a partir das simulações com cinco usuários de voz, esse parâmetro passa a ter um valor regular e não sofre grandes impactos diante do aumento de usuários de voz. Esse comportamento ocorre porque o aumento de usuários de voz faz com que no sistema haja uma maior quantidade de pacotes deste tipo de tráfego. Como os pacotes de voz possuem tamanho inferior aos pacotes de dados, terá menor tempo de serviço e consequentemente menor tempo de sistema.

Utilizando o algoritmo FIFO, há uma queda no tempo geral de fila e de sistema entre as simulações com dois e quatro usuários. Porém, esses tempos aumentam gradativamente a partir

das simulações com cinco usuários. Este mesmo comportamento ocorre com os tempos de sistema dos tráfegos de voz e dados. Entretanto em todos os casos simulados neste cenário, os tempos de sistema estiveram entre 0,001 s e 0,0020 s, respectivamente com quatro e 15 usuários de voz no sistema.

Comparando os escalonadores, no que se refere ao Cenário 2, percebe-se que no escalonador DRR há um decréscimo no tempo de sistema até que o mesmo entre em equilíbrio. Já o FIFO acarreta um aumento singular com o incremento de usuários de voz no sistema.

No terceiro cenário foram feitas simulações aumentando a quantidade de usuários de vídeo de um até seis usuários, enquanto foi mantida a quantidade de quatro usuários de dados em todas as simulações. Então, a Figura 4.13 apresenta as simulações referentes ao terceiro cenário, utilizando-se a Fonte 1.0 e Fonte 2.0 para gerar os pacotes de dados e a Fonte Fixa para a geração do tráfego de vídeo, sendo analisado o comportamento do escalonador DRR.

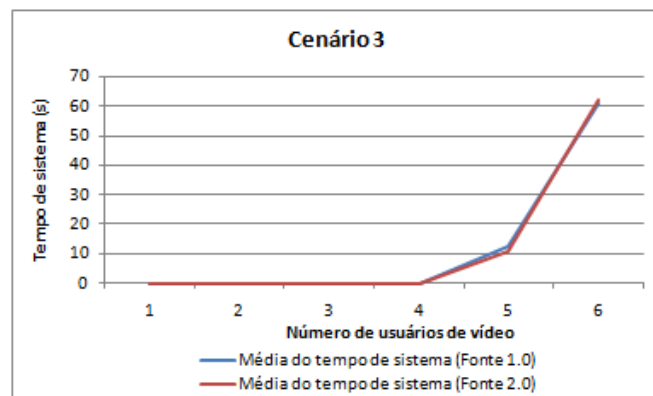


Figura 4.13: Simulação com aumento de usuários de vídeo e quantidade fixa de quatro usuários de dados no escalonador DRR

De acordo com a Figura 4.13, observa-se que a inserção de usuários de vídeo provoca o aumento dos valores do tempo de fila e de sistema. Porém esse aumento é mais significativo a partir da simulação com quatro usuários de vídeo.

É importante observar também que os resultados das simulações com a Fonte 2.0 é semelhante às simulações com a Fonte 1.0, como ocorre também no segundo cenário. No entanto, os

valores do tempo de fila e de sistema possuem maior valor nas simulações com a primeira fonte. Conforme já foi mencionado, essa diferença entre as simulações com as duas fontes ocorre porque com a Fonte 1.0 mais pacotes podem ser enviados em um único ciclo, aumentando o tráfego no escalonador DRR e conseqüentemente aumentando o tempo de fila e de sistema. Já na Fonte 2.0 os pacotes são gerados mais lentamente, produzindo menor tráfego no escalonador e também menor tempo de fila e de sistema.

Porém, é notável que as diferenças dos valores do tempo de sistema entre as simulações com as duas fontes são bem menores no terceiro cenário que no primeiro cenário.

Outro ponto importante a ser observado é que o impacto do aumento de usuários de vídeo é maior do que o impacto causado pelos aumentos dos usuários de dados e de voz. Por exemplo, com cinco usuários, no Cenário 1, o tempo de fila com a Fonte 1.0 teve valor de 0,0101 s, no segundo cenário o valor foi de 0,0008 s, enquanto no terceiro cenário, onde foram aumentados os usuários de vídeo, foi de 12,40 s.

A Figura 4.14 apresenta o comparativo entre os escalonadores FIFO e DRR para o Cenário 3, sendo que a geração do tráfego de vídeo foi realizada com a Fonte Fixa e o tráfego de dados com a Fonte 1.0.

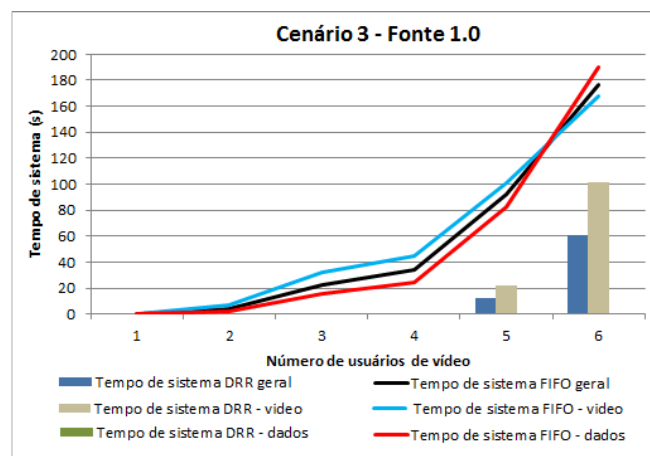


Figura 4.14: Simulações com o aumento de usuários de vídeo e quantidade fixa de quatro usuários de dados

Observa-se, pela Figura 4.14, que utilizando o algoritmo DRR, o aumento de usuários de vídeo provoca um comportamento oposto ao que ocorre diante do aumento de usuários de voz, pois entre um e quatro usuários o sistema permanece estável. A partir das simulações com cinco usuários, há um grande aumento no tempo de sistema das simulações. Por exemplo, até as simulações com cinco usuários, o tempo de sistema geral das simulações foi inferior a 0,01 s. Já na simulação com seis usuários de vídeo, o valor desse parâmetro foi de 60,76 s.

Outro ponto importante a ser observado é que, utilizando o algoritmo DRR, o aumento dos usuários de vídeo não afeta o valor do tempo de sistema do tráfego de dados, que em todas as simulações é inferior a 0,005 s (Na Figura 4.13, devido a escala utilizada, esses valores não são perceptíveis, mas estão todos junto ao eixo da abscissa). Isso ocorre porque existe uma quantidade bem superior de pacotes de vídeo no sistema e também porque em todas as simulações existe a quantidade fixa de quatro usuários de dados.

Também deve ser observado que ao contrário do que ocorre com o DRR, no FIFO, a partir de dois usuários de vídeo, os tempos de sistema começam a aumentar. Porém o tempo de sistema aumenta consideravelmente a partir de três usuários de vídeo. É importante mencionar que com seis usuários de vídeo, usando o algoritmo FIFO, o tempo de sistema é de 176,67 s. Esse valor significa que o sistema já está na região instável e o tempo de sistema crescerá indefinidamente com o aumento do número de usuários.

No Cenário 4 foram feitas simulações aumentando o número de usuários de vídeo de um até seis usuários, enquanto foi mantida a quantidade de quatro usuários de dados e cinco usuários de voz. Optou-se por fixar em cinco usuários de voz, pois como se observa nas Figuras 4.11 e 4.12, com cinco usuários de voz o tempo de sistema já está em equilíbrio estatístico, sem aumento considerável no tempo de sistema. Desta forma, neste cenário é possível observar melhor a influência do tráfego de vídeo.

Assim, a Figura 4.15 apresenta as simulações do quarto cenário, utilizando-se as Fontes 1.0 e 2.0 para gerar os pacotes de dados, e a Fonte Fixa para os tráfegos de voz e vídeo. Vale salientar que apenas os resultados para o escalonador DRR serão apresentados.

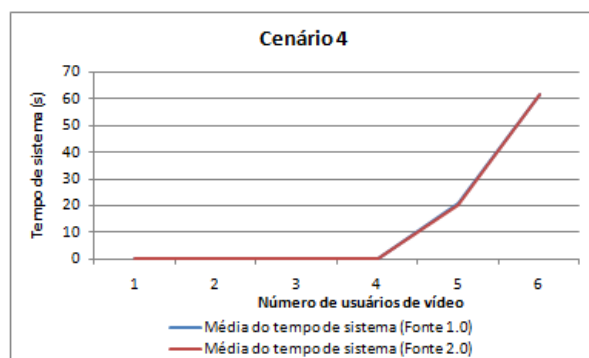


Figura 4.15: Simulação com aumento de usuários de vídeo e quantidade fixa de usuários de voz e de dados no escalonador DRR

Através da Figura 4.15 percebe-se que o aumento de usuários de vídeo, enquanto a quantidade de usuários de voz e dados é fixa, provoca o aumento dos tempos de fila e de sistema. Por exemplo, com um usuário, nas simulações com a Fonte 1.0, o tempo de fila foi de 0,0021 s, enquanto que com seis usuários esse valor foi de 61,49 s.

Assim como ocorreram no segundo e terceiro cenários, pode ser observado também que as simulações com a Fonte 2.0 possuem comportamento semelhante às simulações com a Fonte 1.0. Além disso, conforme ocorreu no Cenário 3, os valores do tempo de fila e de sistema tiveram menores valores nas simulações com a Fonte 2.0 do que com a Fonte 1.0.

A Figura 4.16 apresenta o comparativo entre as simulações realizadas com o escalonador FIFO e DRR para o Cenário 4.

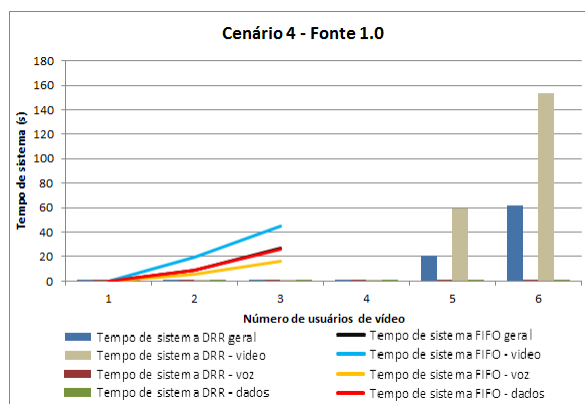


Figura 4.16: Simulações com o aumento de usuários de vídeo e quantidade fixa de cinco usuários de voz e de quatro usuários de dados

Observa-se pela Figura 4.16 que o aumento de usuários de vídeo, no escalonador DRR, não provoca grandes impactos nem no tráfego de voz e nem no tráfego de dados, já que o tempo de sistema de ambos é inferior a 0,01 s em todas as simulações.

Porém, observa-se que a inclusão de usuários de voz, com o escalonador DRR, provocou um aumento no tempo de sistema do tráfego de vídeo. Isso pode ser notado comparando com o Cenário 3, que não possui usuários de voz, e que o tempo de sistema de vídeo foi de 101,35 s com seis usuários. Já, no Cenário 4, com a mesma quantidade de usuários de vídeo, o tempo de sistema de vídeo foi bem superior, tendo o valor de 153,88 s.

Além disso, no Cenário 4, com o escalonador DRR, o aumento do tempo de sistema geral e do tempo de sistema do tráfego de vídeo é mais acentuado a partir de cinco usuários de vídeo, enquanto que no Cenário 3 o aumento só ocorreu com maior intensidade a partir de cinco usuários de vídeo.

Já com o algoritmo FIFO, os tempos de sistema aumentaram gradativamente, conforme o aumento dos usuários de vídeo. Observa-se, também, que o tempo de sistema de vídeo apresentado nas simulações com três usuários já superava o tempo de sistema apresentado na simulação com quatro usuários de vídeo utilizando o DRR. Além disso, do mesmo modo como ocorreu no Cenário 3, o aumento na quantidade de usuários de vídeo, influenciou no aumento dos tempos de sistema dos demais tráfegos, o que não acontece quando os usuários de voz aumentam, como pode ser visualizado no Cenário 2.

No Cenário 5 foram feitas simulações aumentando gradativamente a quantidade de usuários de voz até dezesseis usuários, enquanto foi mantida a quantidade de quatro usuários de dados e um usuário de vídeo em todas as simulações. Os tráfegos de voz e vídeo foram gerados com a Fonte Fixa. Diante disso, a Figura 4.17 apresenta as simulações do quinto cenário, utilizando a Fonte 1.0 e Fonte 2.0 para a geração dos pacotes de dados no escalonador DRR.

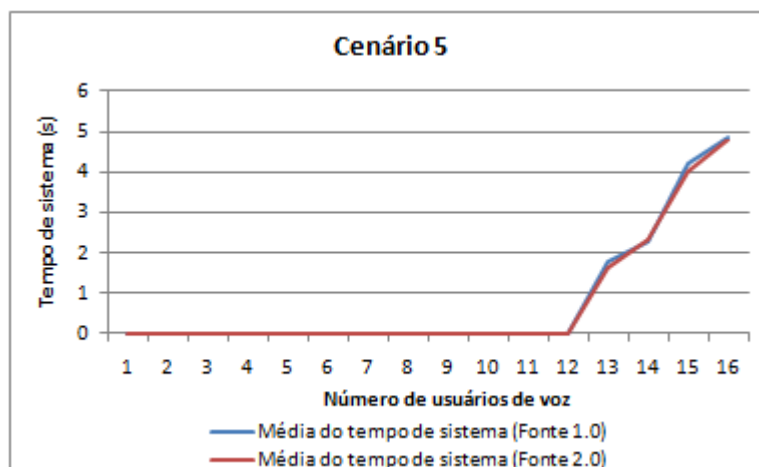


Figura 4.17: Simulação com o aumento de usuários de voz e quantidade fixa de usuários de vídeo e de dados no escalonador DRR

Diante do que é apresentado pela Figura 4.17, observa-se que o aumento dos usuários de voz, provoca o aumento dos tempos de sistema. Porém esse aumento é mais intenso a partir das simulações com doze usuários. Por exemplo, a diferença entre o tempo de fila do primeiro usuário em relação ao décimo segundo usuário, nas simulações com a Fonte 1.0, é de 0,0024 s. Esse valor é muito inferior a diferença entre o tempo de fila do décimo segundo usuário e décimo terceiro usuário, que é 1,78 s.

Nas simulações com a Fonte 2.0, o aumento de usuários de voz também provoca aumento nos tempos de sistema das simulações. Porém, esse aumento é maior nas simulações de um até nove usuários com esta fonte. Contudo, a partir das simulações com nove usuários, o tempo de sistema da Fonte 1.0 passa a ser maior.

Desta forma, é possível afirmar que quando o enlace físico está próximo à região de saturação, um pequeno incremento de tráfego pode congestionar o sistema.

O comparativo entre os escalonadores FIFO e DRR, considerando o Cenário 5 é apresentado na Figura 4.18.

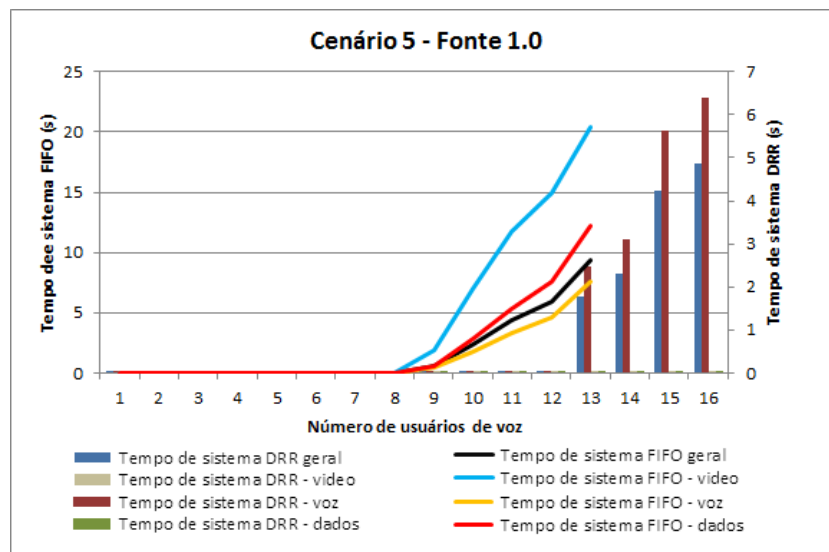


Figura 4.18: Simulações com o aumento de usuários de voz e quantidade fixa de um usuário de vídeo e de quatro usuários de dados

Conforme pode ser observado pela Figura 4.18, com o DRR, inicialmente a média do tempo de sistema sofre um insignificante decréscimo com a inserção de usuários de voz. Isso ocorre até sete usuários de voz. A partir de sete usuários, o tempo de sistema sofre um singelo incremento, até doze usuários de voz no sistema. A partir deste instante, como o sistema já está instável, há um aumento brusco na média do tempo de sistema. É importante observar que apesar do aumento no tempo de sistema geral e de voz, as médias do tempo de sistema de vídeo e de dados não sofreram mudanças expressivas.

Neste aspecto, com o acréscimo de um usuário de vídeo neste cenário, o sistema se comportou de modo similar ao Cenário 2, que só possui tráfego de dados e voz, até doze usuários de voz. Isso comprova que quando o sistema já está próximo da saturação, ínfimos incrementos de tráfegos já acarretam mudanças significativas, o que também ocorreu nos Cenários 1, 3 e 4.

Mediante os fatos, pode-se concluir que a inclusão de usuários de voz no sistema contribui para a melhoria do desempenho dos tempos de sistema dos demais tráfegos, no que se refere ao escalonador DRR. Além disso, conclui-se, também, que o sistema pode aceitar uma quantidade maior de usuários de voz do que de vídeo e dados.

Utilizando o algoritmo FIFO, os tempos de sistema de todos os tráfegos aumentam a partir das simulações com seis usuários de voz. Contudo, este aumento é mais expressivo após a inserção do oitavo usuário de voz. Com oito usuários de voz, os valores relativos ao tempo de sistema para o tráfego de vídeo são muito superiores ao maior tempo de sistema apresentado pelo algoritmo DRR. Além disso, com o FIFO, na simulação com onze usuários de voz, o tempo geral de sistema foi de 4,38 s e o tempo de sistema de vídeo foi de 11,77 s. Com o DRR, estes tempos são respectivamente 0,0024 s e 0,0037 s. Outro fator importante é que os escalonadores FIFO e DRR tem comportamento similares no que se refere ao aumento repentino no tempo de sistema, ou seja, quando este já está próximo a saturação. Porém, o DRR admite um maior número de usuários no sistema.

O sexto cenário é semelhante ao cenário anterior, pois foram feitas simulações aumentando-se a quantidade de usuários de voz de um até quatorze usuários, enquanto foi mantida a quantidade de quatro usuários de dados. Porém, os dois cenários são diferentes em relação à quantidade de usuários de vídeo. No Cenário 5 foram feitas simulações com a quantidade fixa de um usuário de vídeo, enquanto que neste cenário foram considerados dois usuários de vídeo em todas as simulações. O intuito é analisar o impacto provocado pelo aumento de usuários de vídeo. Diante disso, a Figura 4.19 apresenta as simulações referentes ao Cenário 6, utilizando a Fonte 1.0 e a Fonte 2.0 para a geração dos pacotes de dados e a Fonte Fixa para gerar os tráfegos de voz e vídeo. Isso considerando o escalonador DRR.

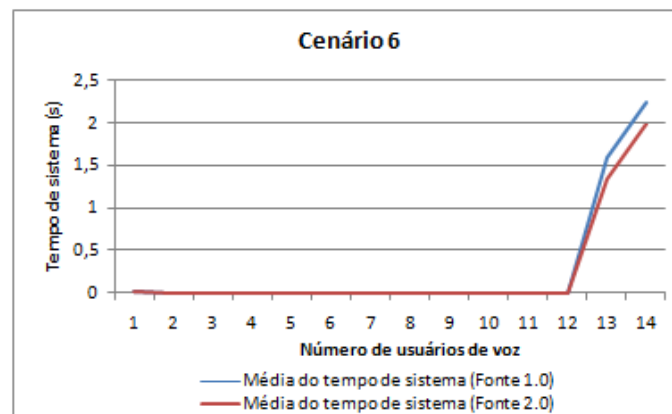


Figura 4.19: Simulação com aumento de usuários de voz e quantidade fixa de dois usuários de vídeo e de quatro usuários de dados no escalonador DRR

Conforme pode ser observado pela Figura 4.19, o aumento de um usuário de vídeo nas simulações do Cenário 6, não provocou grande diferença em relação aos resultados das simulações do Cenário 5. Contudo, em ambos os cenários, a partir do décimo segundo usuário no sistema há um aumento considerável no tempo de sistema.

Comparando as simulações com a Fonte 1.0 em relação às simulações com a Fonte 2.0, no Cenário 5, observa-se, através da Figura 4.17, que os tempos de fila são menores com a segunda fonte. Logo, neste último cenário, o aumento de usuários de voz provoca maior impacto nas simulações com a primeira fonte de tráfego.

O comparativo entre os escalonadores FIFO e DRR, considerando o Cenário 6 é demonstrado na Figura 4.20.

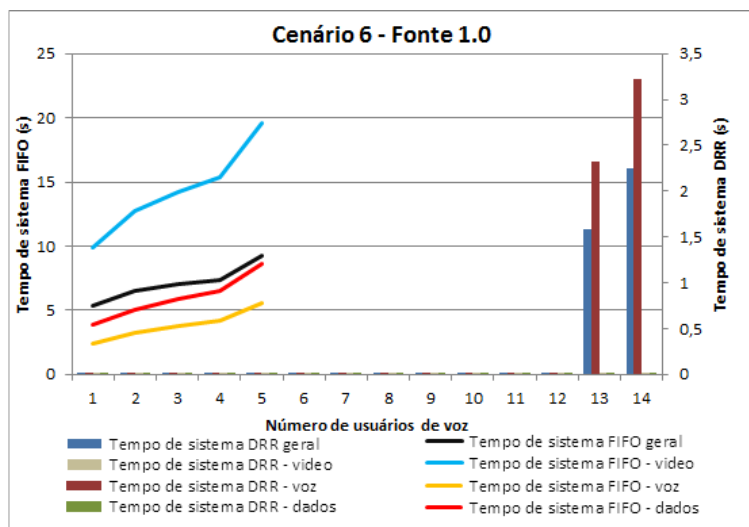


Figura 4.20: Simulações com o aumento de usuários de voz e quantidade fixa de dois usuários de vídeo e quatro usuários de dados

Observa-se, pela Figura 4.20, que com o escalonador DRR, o aumento de um usuário de vídeo no Cenário 6 não altera o comportamento do sistema em relação ao que foi analisado no Cenário 5, pois da mesma forma que no cenário anterior, houve inicialmente a queda do tempo de sistema de todos os tráfegos até sete usuários de voz. A partir do sétimo usuário de voz, os tempos de sistema geral e de voz aumentaram sem significativas alterações até doze usuários de voz. No entanto, com treze usuários de voz, como o sistema está próximo da saturação, o aumento nos

tempos de sistema geral e de voz é mais expressivo. Contudo, os tempos de sistema de vídeo e dados foram constantes em todas as simulações.

Analisando o escalonador FIFO, como ele não permite priorizar nenhum tráfego, com o incremento de um usuário de vídeo o tempo de sistema geral com um único usuário de voz já é de 5 s, sendo o tempo de sistema de vídeo de 9 s. Estes valores são crescentes com o aumento de um único usuário de voz, que dentre os tráfegos, possui o menor pacote. Isso demonstra que quando o sistema já está próximo a sua região de saturação, um único usuário compromete o sistema.

Outra análise entre os escalonadores FIFO e DRR foi realizada. Desta vez, o objetivo foi verificar o impacto dos parâmetros e formas de geração das fontes nestes escalonadores. Para isso, simulações com parâmetros diferentes dos mencionados na Tabela 3.1, apresentada no Capítulo 3, foram realizadas. As simulações usaram a mesma fonte para a geração dos tráfegos de voz, vídeo e dados, porém com valores referentes ao tamanho dos pacotes, tempo dos intervalos Off e taxa de pico diferentes. Para tanto, dois cenários para cada um dos tráfegos foi analisado. O intuito é verificar o impacto que as Fontes Fixa, Fonte 1.0 e Fonte 2.0 provocam nos escalonadores FIFO e DRR analisando-se o tempo de fila, tempo de serviço, tempo de sistema, bem como a quantidade de pacotes descartados.

Deste modo, foram feitas simulações nos dois simuladores, com 300000 pacotes no sistema, sendo 100000 para cada um dos tráfegos. Vale salientar que com 100000 pacotes para cada tipo de tráfego os escalonadores já estavam convergindo. Nestas simulações, os valores referentes ao tamanho dos pacotes e o tempo dos intervalos On e Off foram alterados, permitindo verificar o seu impacto nos sistemas computacionais.

No simulador FIFO, testes foram realizados alterando o tamanho do buffer e a capacidade do canal. Então, pode-se verificar que o tamanho dos pacotes exerce grande influência no desempenho dos processadores e nós da rede. Porém, se a capacidade do canal sofrer alterações nas mesmas proporções, os resultados serão similares. Isso se justifica pelo fato que com pacotes maiores, haverá maior morosidade no tempo de processamento, o que irá aumentar o tempo de fila e também o tempo no sistema.

Também foi possível constatar que o tamanho da fila de espera não influencia no tempo de serviço, mas tem repercussão imediata no número de pacotes descartados. Contudo, com uma fila superior a 150 posições, não há perdas de pacote se o link for superior ou igual a 1 Mb.

Já, no simulador DRR, testes foram realizados alterando-se o tamanho da quota, o tamanho do buffer e a capacidade do canal. Então, com os resultados obtidos, pode-se verificar que o parâmetro mais influente neste escalonador é a quota. Foi, também, observado que o tamanho da quota não pode ser igual para os tráfegos de voz, vídeo e dados, pois, desta forma, o primeiro tráfego é beneficiado, já que possui menor tamanho, e os dois últimos são prejudicados, em relação ao tempo de fila, tempo de serviço e descarte de pacotes, que serão maiores. Então, para haver maior justiça, a quota deve ser proporcional ao tamanho dos pacotes.

Outro ponto importante que foi observado, é que os modelos de fonte propostos provocam impactos diferentes em relação aos dois escalonadores. No FIFO, a fonte que causa maior tempo de fila é a fonte On/Off Variável 1.0, que consequentemente provoca maiores tempos de sistema e maior descarte de pacotes. Já no DRR, o maior tempo de fila é causado pela fonte On/Off Variável 2.0, que também implica em maiores tempos de sistema e maior descarte de pacotes. Diante disso, pode-se notar que no escalonador FIFO o impacto causado pelas fontes ocorre, principalmente, em função do tempo de chegada das mesmas, além da alteração nos parâmetros necessários para a sua geração. Desta forma, a fonte On/Off Fixa que gera tempos de chegada menores, provoca maior descarte no escalonador FIFO do que no DRR. Por outro lado, no escalonador DRR, o impacto causado pelas fontes ocorre, principalmente, em relação à média do tamanho dos pacotes gerados, devido à distribuição de quantum para as filas. Logo, a fonte On/Off Variável 2.0 gera maior descarte de pacotes no escalonador DRR, pois gera pacotes com tamanho médio superior aos das outras duas fontes.

Desta forma, pode-se perceber que as fontes exercem muita influência nos processadores e nós da rede, sobretudo no que tange ao tamanho dos pacotes e taxa de geração dos mesmos. Assim, dependendo da QoS almejada, um determinado tipo de fonte e escalonador deve ser usado.

Para reforçar ainda mais a diferença que o quantum exerce no escalonador DRR, simulações com quantum de 5440 bits para o tráfego de voz, 8192 bits para o tráfego de vídeo e de 4096 bits para o tráfego de dados foram realizadas. Nestas, os mesmos seis cenários foram considerados, bem como os parâmetros para a geração das fontes apresentados na Tabela 3.1 foram usados.

Mediante a análise dos dados, constatou-se que nos Cenários 1, 2, 3 e 4, não houve mudanças expressivas das analisadas com quantum inferior para os tráfegos de voz e vídeo. No entanto, para o Cenário 5 e 6, onde há um aumento de usuários de voz enquanto os usuários de vídeo e dados são fixos, o comportamento teve variações consideráveis, sobretudo no Cenário 5.

Na verdade, como nestas simulações o quantum de voz é aproximadamente quatro vezes maior que o tamanho dos pacotes de voz, estes são priorizados. Assim, no Cenário 5 e 6, com o quantum de 5440 bits, o comportamento do sistema é similar ao que ocorre no Cenário 2, com quantum de 2720 bits, apresentado nas Figuras 4.11 e 4.12, ou seja, o sistema se estabiliza e não chega a sua saturação, mesmo com dezesseis usuários de voz no sistema.

4.2.2 Conclusões Referentes ao Escalonador DRR Confrontado com o Escalonador FIFO

Neste tópico foi analisado o impacto que as aplicações multimídia geram nas redes atuais, caso os recursos sejam mantidos. Desta forma, seis cenários foram investigados, sendo que o primeiro se restringe ao tráfego de dados, que era uma realidade até recentemente. Nos demais cenários, fez-se simulações aumentando gradativamente os tráfegos de voz, vídeo ou ambos. O intuito foi verificar as mudanças no desempenho de um nó de rede de 2 Mbps, bem como comparar a performance dos escalonadores FIFO e DRR, quando os tráfegos de voz e vídeo são introduzidos. Desta forma, as plataformas de simulação desenvolvidas contemplam um buffer de 1.000.000 posições, um link de 2 Mbps, as fontes geradoras de tráfego propostas e os algoritmos de escalonamento.

Além disso, como forma de verificar como a quota influencia no desempenho do escalonador DRR, bem como permite atribuir prioridades, diferentes quotas foram utilizadas nos tráfegos de voz e vídeo. Isso foi feito devido aos tráfegos de voz e vídeo requererem maior QoS por parte dos usuários. Assim, foi possível visualizar que priorizando os serviços de voz e vídeo, um maior número de usuários no sistema é permitido.

Como o buffer do nó de rede analisado é de 1.000.000 de posições, o que praticamente representa um buffer infinito, nenhuma perda de pacotes ocorreu. Contudo, aumentando o número de usuários de voz, vídeo e dados, os tempos de sistema e de fila aumentam significativamente no escalonador DRR a partir de um determinado número de usuários. A única exceção ao fato ocorreu no segundo cenário, onde o tempo de sistema praticamente se estabilizou independente do aumento no número de usuários. É importante mencionar que neste cenário, mesmo com dezesseis usuários de voz, o sistema não ficou instável. Como observado nas Figuras 4.3, 4.11 e 4.12, onde foram introduzidos gradativamente os usuários de voz, restringindo-se ao máximo de quinze usuários deste tráfego, o que representa um volume de 1.050.000 pacotes na rede no Cenário 6, percebe-se que o escalonador DRR admite um número considerável de usuários de voz no sistema. Já com o FIFO, o incremento de qualquer usuário, causa o aumento do tempo de sistema e de fila.

Outrossim, apenas os resultados da geração do tráfego de dados com a Fonte 1.0 foram apresentados nos gráficos comparativos, pois os resultados são muito similares aos obtidos com a Fonte 2.0. Comparando os escalonadores FIFO e DRR, pode-se verificar que neste último foi possível inserir um maior número de usuários de voz e vídeo, conforme pode ser comprovado através dos Cenários 4, 5 e 6. Porém, para os tráfegos de dados os escalonadores FIFO e DRR tem desempenho similar, sendo que o FIFO é um pouquinho melhor. Contudo, como o número de aplicações multimídia cresce vertiginosamente, alterar o algoritmo de escalonamento dos nós da rede é uma boa solução para a melhoria no desempenho dos sistemas. Além disso, a diferença entre as Fontes 1.0 e 2.0 é mais expressiva no escalonador FIFO, em relação ao comportamento do tempo de sistema diante da variação de usuários nas simulações realizadas.

Ainda analisando as simulações com o escalonador FIFO e DRR com relação ao primeiro cenário, observa-se que o aumento de usuários de dados provocou saturação no sistema com sete usuários, em relação à Fonte 1.0, nos dois escalonadores. No entanto, em relação às simulações com a Fonte 2.0, a saturação do sistema acontece a partir das simulações com sete usuários de dados, no escalonador DRR, enquanto que no escalonador FIFO, ocorre a partir das simulações com oito usuários de dados. Outro ponto importante, é que o impacto do aumento de usuários de dados é maior no escalonador DRR do que no FIFO. Por exemplo, no Cenário 1, nas simulações com nove usuários e Fonte 2.0, o valor do tempo de sistema no escalonador DRR foi superior à 200 s, enquanto que no FIFO, esse valor foi inferior à 100 s.

Comparando-se os dois escalonadores em relação ao Cenário 2, observa-se que o aumento de usuários de voz, enquanto a quantidade de usuário de dados permanece fixa, torna o sistema estável, com baixa variação do tempo de sistema em ambos os escalonadores. Deste modo não há degradação da qualidade de serviço. Outro fato relevante é que o aumento de um único pacote de voz, se o sistema já está próximo da saturação, aumenta bruscamente o tempo do sistema, o que pode ser comprovado através do quinto e sexto cenários. Esta é outra situação que também ocorre no escalonador FIFO, sendo bem enfática no Cenário 6.

Por outro lado, o comportamento das simulações com os dois escalonadores é semelhante no Cenário 3, já que em ambos, o aumento de usuários de vídeo provocou o aumento do tempo de sistema. No entanto, observou-se que esse aumento é muito superior em relação ao escalonador FIFO. Por exemplo, com seis usuários de vídeo e Fonte 1.0, o tempo de sistema no escalonador FIFO foi superior a 170 s, enquanto que no escalonador DRR, o valor desse parâmetro foi inferior a 70 s. Outro ponto importante é que no escalonador FIFO, as diferenças entre os resultados das simulações com as Fonte 1.0 e Fonte 2.0 são bem superiores ao que ocorre no escalonador DRR. Diante disso, pode-se concluir que o aumento de usuários de vídeo provoca maior impacto no escalonador FIFO que no DRR. Além disso, é evidenciado através das Figuras 4.13 e 4.14 que com o escalonador DRR admite um maior número de usuários de vídeo.

Em relação ao quarto cenário, observa-se que o aumento de usuários de vídeo, enquanto a quantidade de usuários de voz e dados permaneceu fixa, provocou grandes diferenças entre as simulações com a Fonte 1.0 e 2.0 no escalonador FIFO, sendo que o aumento do tempo de sistema foi bem superior com a primeira fonte. Porém, o mesmo não ocorreu com o escalonador DRR, pois os resultados e o comportamento das simulações com as duas fontes foram semelhantes. Além disso, observou-se que o aumento do tempo de sistema foi bem superior nas simulações com o escalonador FIFO do que no DRR, sendo que com três usuários de vídeo, esse parâmetro no primeiro escalonador possuía um valor superior a 25 s com a Fonte 1.0, enquanto que no segundo escalonador esse valor foi inferior a 2 s.

Em relação ao quinto e sexto cenário, observou-se que o impacto do aumento de usuário de voz, enquanto manteve-se a quantidade de usuários de vídeo e dados fixa, provocou maiores impactos no escalonador FIFO do que no DRR. Por exemplo, nos dois cenários, nas simulações com o escalonador FIFO, o valor do tempo de sistema passou a aumentar a partir das simulações com quatro usuários de voz, enquanto que no escalonador DRR, esse aumento foi observado apenas a partir das simulações com doze usuários. Outro ponto importante a ser observado é que foi aumentado um usuário de vídeo no sexto cenário, em relação ao quinto cenário. Esse aumento provocou grandes diferenças em relação aos resultados entre os dois cenários, sendo que houve grande aumento do tempo de sistema no Cenário 6, no escalonador FIFO, enquanto que no DRR houve grande semelhança entre os resultados.

Desta forma, pode-se concluir que o escalonador DRR produz melhores resultados que o escalonador FIFO, na maioria das situações de utilização das redes. Conforme foi mencionado, o escalonador FIFO só foi melhor que o escalonador DRR, no primeiro cenário no qual há apenas tráfego de dados no sistema, sendo aumentado gradativamente o número de usuários. Além disso, observou-se também que a mudança na fonte de tráfego, só provoca grandes impactos no escalonador FIFO, com melhores resultados nas simulações com a Fonte 2.0.

Também se constata que quando há recurso limitado do canal, e um determinado número de usuários para serem atendidos, os tipos de serviço devem ser definidos e limitados. Assim, dependendo da QoS requerida, certos tipos de serviço não podem ser contratados.

Além disso, comprovou-se que o escalonador DRR admite um maior número de usuários de voz e vídeo no sistema. Contudo, esta premissa não é verdade no que se refere ao tráfego de dados. No entanto, isso pode ser decorrente do valor do quantum atribuído. Desta forma, conclui-se que a alteração do algoritmo de escalonamento deve ser realizada. Se a substituição for feita para o algoritmo DRR, é necessário analisar os Acordos de Níveis de Serviço (SLA) e identificar qual o perfil dos usuários de tal forma a atribuir quotas adequadas para cada tráfego, atendendo os usuários com a QoS contratada.

Tendo em vista a influência que a quota exerce e visando distribuir a largura de banda de forma a manter a equidade entre os tráfegos, propõe-se a utilização do escalonador com distribuição de quota baseada na banda efetiva. Este algoritmo de escalonamento permite minimizar as perdas de pacotes, sendo um dos requisitos necessários no que se refere à qualidade dos serviços prestados. Outrossim, o escalonador usa o conceito de largura de banda efetiva ou equivalente, que consiste em uma largura de banda fixa estimada através dos parâmetros definidos nos Acordos de Níveis de Serviço (SLA), para a distribuição do quantum, sendo este o próximo escalonador analisado.

4.3 ANÁLISE COM O ESCALONADOR BASEADO EM BANDA EFETIVA

Neste tópico será analisado o desempenho do escalonador inédito, que usa o conceito de banda efetiva para a distribuição das quotas entre os tráfegos. Isso permite atender aos Acordos de Níveis de Serviço (SLA) das aplicações, sobretudo no que se refere à perda de pacotes.

Para estudar essa eficiência foi desenvolvida uma plataforma de simulação usando o software MatLab, e o modelo de banda efetiva é baseado na fórmula apresentada em [62]. A

fórmula utilizada em [62], por sua vez, utiliza fontes On/Off. A Tabela 4.3 mostra os parâmetros de fonte On/Off utilizados.

Tabela 4.3: Taxa de pico, tamanho dos pacotes e intervalos de On e Off das fontes de dados, voz e vídeo.

Tipos de Tráfego	Taxa de pico (bps)	Tamanho do pacote (bits)	Intervalos de Ton e Toff (s)
Dados	307200	4096/8192	0,003
Voz	65536	1360	0,0016
Vídeo	2097152	2048/4096	0,0010

Como se observa pela Tabela 4.3, somente três tipos de fontes foram consideradas: voz, dados e vídeo. Dois diferentes tamanhos de pacotes são usados para dados e vídeo para estudar o efeito da perda de pacotes e do tempo de espera dos pacotes. Além disso, é adotado o mesmo tempo médio de On e Off para todos os tráfegos.

Se forem comparados os valores usados nesta simulação com os apresentados na Tabela 3.1, será possível perceber que as diferenças são poucas. Os tempos dos intervalos On e Off são os mesmos e quanto a taxa de pico, apenas a de vídeo é o dobro da apresentada na Tabela 3.1. Consoante ao tamanho dos pacotes, para testar o escalonador proposto, os tráfegos de vídeo e dados usam dois tamanhos distintos.

4.3.1 Resultados dos Cenários Simulados com o Escalonador com Banda Efetiva

Nesta seção apresentam-se os resultados obtidos através do simulador desenvolvido em MatLab 7.0 para escalonamento de tráfego utilizando a função da banda efetiva proposta em [62]. Dois cenários são analisados: o primeiro em que a quota é a mesma para todos os tipos de tráfego e o segundo em que a quota definida para cada tráfego é proporcional à sua banda efetiva. Além disso, para cada cenário são utilizados buffers de tamanho finito e infinito. O fluxograma do simulador desenvolvido é o mesmo apresentado na Figura 2.2.

Para validar as simulações, executou-se cinco vezes o aplicativo desenvolvido para cada situação, permitindo maior segurança e confiança nos resultados apresentados, sendo a média demonstrada nas tabelas a seguir. Isso foi realizado para identificar o intervalo de confiança do sistema.

Além disso, as simulações foram realizadas considerando 2.000 pacotes no sistema para cada tipo de tráfego, descartando-se o regime transitório. Sendo assim, as análises compreendem o intervalo entre 500 e 2.000 pacotes para cada tráfego.

Para todas as simulações executadas, considerou-se a capacidade do enlace de 2 Mbps e a taxa de perda equivalente a 10^{-3} para todos os tipos de tráfego.

Nas tabelas a seguir são apresentados os resultados obtidos nas simulações. As siglas utilizadas nas tabelas têm os seguintes significados: TMP, tamanho médio dos pacotes; $E\{X\}$, tempo médio de serviço; $E\{W\}$, tempo médio de espera no buffer; $E\{T\}$, tempo médio no sistema, equivalente a $E\{W\} + E\{X\}$ e PP, perda de pacotes.

As Tabelas 4.4 e 4.5 mostram os resultados obtidos para o escalonador DRR com quota fixa de 512 bits para cada tráfego. O tamanho de buffer considerado é de 500 posições.

Tabela 4.4: Resultados obtidos com o escalonador DRR com quota fixa de 512 bits, buffer finito e com pacotes de tamanhos maiores.

Tipos de Tráfego / Parâmetros	Tráfego de Voz	Tráfego de Vídeo	Tráfego de Dados
TMP (bits)	!Divisão por zero	!Divisão por zero	!Divisão por zero
T_{On} (segundos)	0,02	0,00,0	0,00,0
T_{Off} (segundos)	0,00,02	0,01	0,00,0
$E\{X\}$ (segundos)	0,00	0,00	0,00
$E\{W\}$ (segundos)	0,00	541,92	0,0
$E\{T\}$ (segundos)	0,00	541,92	84,46
PP	0,0	855,00	1408,20

Tabela 4.5: Resultados obtidos com o escalonador DRR com quota fixa de 512 bits, buffer finito e com pacotes de tamanhos menores.

Tipos de Tráfego / Parâmetros	Tráfego de Voz	Tráfego de Vídeo	Tráfego de Dados
TMP (bits)	1361,42	1999,41	4104,40
T_{On} (segundos)	0,02	0,01	0,03
T_{Off} (segundos)	0,02	0,01	0,03
E{X} (segundos)	0,00	0,00	0,00
E{W} (segundos)	0,00	680,52	278,29
E{T} (segundos)	0,00	680,53	278,29
PP	0,00	212,00	1053,40

Como pode ser observada pelas Tabelas 4.4 e 4.5, a perda de pacotes referente ao tráfego de voz não ocorre. Isso pode ser justificado em função do tamanho do pacote que é menor, conduzindo para menores tempos de serviço e reduzindo o tempo no sistema. Outro fato que pode ser observado é que reduzindo o tamanho dos pacotes de vídeo à metade, a taxa de perda diminui quatro vezes. No entanto, quanto ao tráfego de dados, a redução no tamanho do pacote pela metade repercutiu num aumento de quatro vezes no tempo no sistema.

Nas Tabelas 4.6 e 4.7 são apresentados os resultados obtidos com o escalonador DRR com quota proporcional à banda efetiva baseada na fórmula apresentada em [62] para dois tamanhos de pacotes de vídeo e dados. Para este cálculo, foram considerados buffer na saída de 500 posições, capacidade do enlace de 2 Mbps e taxa de perda de 10^{-3} para todos os tipos de tráfego. No entanto, o T_{On}, T_{Off} e a taxa de geração dos pacotes são obtidas pelos valores gerados na simulação. Isso faz diferença nos valores da banda efetiva.

Tabela 4.6: Resultados obtidos considerando o escalonador DRR com quota proporcional à banda efetiva com buffer finito, com tamanhos maiores de pacote.

Tipos de Tráfego / Parâmetros	Tráfego de Voz	Tráfego de Vídeo	Tráfego de Dados
TMP (bits)	1360,34	4027,34	7919,48
T_{On} (segundos)	0,02	0,01	0,03
T_{Off} (segundos)	0,02	0,01	0,03
E{X} (segundos)	0,00	0,00	0,01
E{W} (segundos)	0,00	44,55	516,26
E{T} (segundos)	0,00	44,55	516,26
PP	0,00	0,00	0,00
Banda Efetiva (bps)	194,34	588,07	1278,87

Tabela 4.7: Resultados obtidos considerando o escalonador DRR com quota proporcional a banda efetiva com buffer finito, com tamanhos menores de pacotes.

Tipos de Tráfego / Parâmetros	Tráfego de Voz	Tráfego de Vídeo	Tráfego de Dados
TMP (bits)	1359,07	2020,77	4057,74
T_{On} (segundos)	0,02	0,01	0,03
T_{Off} (segundos)	0,02	0,01	0,03
E{X} (segundos)	0,00	0,00	0,00
E{W} (segundos)	0,00	65,69	550,41
E{T} (segundos)	0,00	65,69	550,42
PP	0,00	0,00	0,00
Banda Efetiva (bps)	194,16	307,45	736,95

Analisando as Tabelas 4.6 e 4.7 pode-se comprovar que independente do tamanho dos pacotes, não ocorre perda de pacotes para nenhum tipo de tráfego. Isso prova que a função da banda efetiva proposta em [62] satisfaz o requisito de QoS para este parâmetro. No entanto, não há garantia do atraso de pacotes. Deste modo, o tráfego de dados é severamente penalizado devido à carga elevada dos outros tráfegos. Além disso, se for somar o tempo médio no sistema dos três tipos de tráfego, não há muita diferença quanto a alterações no tamanho dos pacotes. Outro ponto que merece destaque é que os atrasos são reduzidos de modo proporcional à sua carga, possibilitando uma distribuição equitativa dos recursos.

Com os pacotes maiores, como é mostrado na Tabela 4.6, a estimação da banda efetiva para o tráfego de vídeo é mais do que o dobro em comparação com pacotes menores, assim o tempo de espera dos pacotes diminui substancialmente. A mesma situação acontece com o tráfego de dados, mas a melhoria no tempo de espera é menos notável porque há um pequeno aumento na estimação da largura de banda efetiva neste caso. Outras simulações foram realizadas que provam que a QoS relacionada à perda de pacotes pode ser satisfeita. Entretanto, uma busca para modelar o tempo de espera do pacote é necessária para garantir a QoS para este parâmetro.

Nas Tabelas 4.8 e 4.9 são mostrados os resultados obtidos usando buffers infinitos. Como o valor infinito de K conduz δ a zero na Eq. 7 (cap. 2), os parâmetros α e β não podem ser calculados,

o que implica que a largura de banda efetiva não pode ser estimada. Assim, os valores de largura de banda efetiva mostrados na Tabela 4.9 são calculados considerando o tamanho do buffer de 500 posições.

Tabela 4.8: Resultados obtidos com o escalonador DRR com quota fixada em 512 bits e buffer infinito, com tamanhos maiores de pacote.

DRR COM QUOTA FIXA DE 512 BITS			
Tipos de Tráfego / Parâmetros	Tráfego de Voz	Tráfego de Vídeo	Tráfego de Dados
TMP (bits)	1360,34	4027,34	7919,48
T _{On} (segundos)	0,02	0,01	0,03
T _{Off} (segundos)	0,02	0,01	0,03
E{X} (segundos)	0,00	0,00	0,01
E{W} (segundos)	0,00	1037,36	894,33
E{T} (segundos)	0,00	1037,37	894,34

Tabela 4.9: Resultados obtidos com o escalonador DRR, sendo a quota proporcional a banda efetiva, buffer infinito, com tamanhos maiores de pacote.

DRR COM QUOTA PROPORCIONAL A BANDA EFETIVA DE KESIDIS			
Tipos de Tráfego / Parâmetros	Tráfego de Voz	Tráfego de Vídeo	Tráfego de Dados
TMP (bits)	1360,34	4045,36	7884,60
T _{On} (segundos)	0,02	0,01	0,03
T _{Off} (segundos)	0,02	0,01	0,03
E{X} (segundos)	0,00	0,00	0,01
E{W} (segundos)	0,00	78,04	527,13
E{T} (segundos)	0,00	78,04	527,14
Banda Efetiva (bps)	16350,08	259453,89	78299,35

Comparando a Tabela 4.8 com a Tabela 4.4, pode-se notar que o tempo médio no sistema com buffer infinito é maior como já era esperado, pois não há perda de pacotes. A mesma conclusão pode ser obtida quando comparada a Tabela 4.9 com a Tabela 4.7.

Quando um valor fixo de quantum é usado no esquema de escalonamento DRR, o que significa a mesma oportunidade para todos os tráfegos, os tráfegos com altas cargas são

penalizados como esperado, (ver Tabelas 4.4 e 4.8). No entanto, usando o esquema de escalonamento proposto com base no valor do quantum proporcional à largura de banda efetiva proposta em [62] um esquema de escalonamento mais justo pode ser obtido. Esta permite uma melhora no que se refere à perda de pacotes, tendo em vista que com um buffer de 500 posições não ocorreu nenhuma.

Assim, pode-se comprovar que dependendo das características do tráfego, o uso de uma classificação criteriosa para cada caso pode auxiliar na diminuição das diferenças, de modo que o usuário não as sinta, ou que sejam menos perceptíveis quando da utilização do aplicativo.

4.3.2 Conclusões Referentes ao Escalonador com Banda Efetiva

Neste tópico um novo esquema de escalonamento de tráfego usando a largura de banda efetiva foi apresentado. No esquema de escalonamento proposto, o serviço de cada tráfego, em um ciclo, é proporcional à largura de banda efetiva. Uma vez que no cálculo da largura de banda efetiva alguns parâmetros de QoS são considerados, o esquema de escalonamento proposto pode garantir a QoS de cada tráfego. Um simulador foi desenvolvido para testar a eficiência do esquema proposto e o cálculo da banda efetiva foi baseado na fórmula proposta em [62]. O parâmetro de QoS usado na fórmula apresentada em [62] se restringe à perda de pacotes. Assim, o esquema de escalonamento poderia ser testado somente para este parâmetro, e os resultados obtidos mostraram que o esquema de escalonamento pode trabalhar satisfatoriamente. No entanto, para incluir outros parâmetros de QoS, tais como tempo de espera e jitter, pesquisas devem ser realizadas para encontrar uma nova fórmula para a largura de banda efetiva ou outra técnica que combine a largura de banda efetiva e os pesos para as filas e que considerem os parâmetros de QoS.

Vale salientar que independente do cenário analisado, o escalonador proposto demonstrou desempenho superior comparado com o escalonador DRR com quota fixa para todos os tipos de

tráfego. Além disso, pode-se mencionar que o mesmo fornece garantias de QoS no que se refere ao parâmetro perda de pacotes. No entanto, o tempo de espera precisa ser melhor investigado.

4.4 CONCLUSÕES DO CAPÍTULO

Neste capítulo, aplicaram-se os conceitos sobre modelos de tráfego e suas características, bem como os três modelos de fontes desenvolvidos e analisados no Capítulo 3 foram incorporados para testar alguns escalonadores. Como descrito no capítulo anterior, uma das fontes tem tamanho fixo de pacotes e, também, tempo ativo fixo, o que ocorre com o tráfego de voz e vídeo, e os outros dois tendo estes parâmetros variando de acordo com a distribuição exponencial. Esta variação permitiu uma aproximação com o cenário real das redes convergentes, bem como verificar o impacto dos tipos de tráfego nos sistemas computacionais e nos escalonadores.

Um simulador referente ao algoritmo de escalonamento FIFO também foi desenvolvido, tendo em vista que o mesmo é o mais usado nos equipamentos de interconexão atuais. Através dele, foi possível analisar o comportamento dos diferentes tráfegos em relação ao tempo de sistema, tempo de fila, tempo de serviço. Observou-se que este escalonador era adequado para aplicações onde o tráfego de dados foi predominante. Isso pode ser comprovado através do Cenário 1, o qual se restringe ao tráfego de dados e que o escalonador FIFO teve uma performance levemente melhor se comparado ao escalonador DRR. Porém, quando tráfegos de voz e vídeo predominam na rede, os escalonadores DRR e a proposta do DRR incorporando o cálculo da banda efetiva para a distribuição das quotas são mais adequados.

Também, pode ser observado que o impacto das fontes é mais significativo no escalonador FIFO que no DRR. Além disso, constatou-se que no FIFO, a fonte que gera uma fila maior é a fonte On/Off Variável 1.0, o que conseqüentemente produz maior tempo de serviço e aumenta o descarte de pacotes. No DRR, a maior fila ocorre com a fonte On/Off Variável 2.0, que também aumenta o tempo de sistema e a perda de pacotes. Assim, pode-se notar que no escalonador FIFO os impactos

causados pelas fontes ocorrem principalmente em função do tempo de chegada e de mudanças nos parâmetros necessários para a sua geração.

Outrossim, foi analisado o comportamento do escalonador DRR considerando os mesmos cenários que foram investigados com o escalonador FIFO. Verificou-se que o escalonador DRR consegue distribuir os recursos de modo a manter a equidade entre os tráfegos, sendo adequado para tráfegos multimídia. Isso pode ser comprovado pelo fato que apenas no Cenário 1, onde o tráfego da rede se restringe a dados, seu desempenho teve uma singela desvantagem quando comparado ao escalonador FIFO. Contudo, comprovou-se que o DRR admite um maior número de usuários de voz e vídeo, além de conseguir priorizar estes tráfegos reduzindo o atraso na rede. Vale salientar que para tráfegos multimídia o parâmetro de QoS atraso é mais importante que a perda de pacotes.

Consoante ao escalonador proposto neste trabalho pode-se perceber que o mesmo consegue evitar a perda de pacotes. Contudo, o atraso, fator sensível nas aplicações multimídias, deve ser melhor investigado tendo em vista que são valores expressivos. Uma alternativa para isso é investigar fórmulas para cálculo da banda efetiva que atendam outros parâmetros além da perda de pacotes. No entanto, como as aplicações de dados não admitem perdas de nenhum bit, a solução é apropriada e pode gerar grandes melhorias no desempenho das redes.

5 ESCALONADORES PARA REDE DE SENSORES

Com a evolução da indústria de componentes eletrônicos, bem como o modo de comunicação entre estes dispositivos, surgiram as Redes de Sensores Sem Fios (RSSF). Estas redes são compostas por dispositivos eletrônicos que efetuam o monitoramento remoto de determinado ambiente ou fenômeno e reportam os dados coletados ao interessado [18], [51].

As RSSF possuem aplicações em diversas áreas, podendo monitorar ambientes, plantações, o corpo humano, entre outras [77]. No entanto, caso seja monitorado o corpo humano, estas redes são denominadas Redes de Sensores Corporais. Assim, em determinadas situações, os nós podem estar em locais de difícil acesso, como internos ao corpo humano, complicando a substituição da bateria dos nós, quando necessário.

Com o avanço da indústria de microeletrônica, os nós das redes de sensores reduziram suas dimensões, sendo que alguns modelos podem ser chamados de nano sensores. Isso fez com que as baterias fossem também reduzidas, diminuindo a capacidade de armazenamento de energia. Assim, os nós devem operar tanto visando cumprir o objetivo para o qual a rede foi proposta, quanto para economizar energia.

Desta forma, utilizar técnicas para economizar energia é de extrema importância, pois mesmo sabendo que as baterias podem ser trocadas, muitas vezes esta troca pode ser dificultada devido ao tamanho e a localização dos nós [78].

Dentre as tarefas realizadas pelos nós, a que mais degrada a bateria é a comunicação [112]. Como o meio de transmissão entre os sensores é o ar, uma vez que mais de um nó comece a transmitir pacotes simultaneamente, haverá colisão e os pacotes deverão ser retransmitidos [18].

Uma forma de evitar estas colisões e prolongar assim a duração da bateria é utilizar mecanismos de controle de acesso ao meio [18]. Estes visam impedir que mais de um nó comece a transmitir pacotes no mesmo instante, evitando as colisões de pacotes [51].

Independente da tecnologia usada, os sensores transmitem as informações a um nó centralizador, denominado sorvedouro, o qual envia os pacotes para um servidor ou rede de telecomunicações, onde são armazenados os dados coletados. Como os pacotes que chegam ao nó centralizador vêm de diversos nós e há somente um canal de saída, se o fluxo de dados for grande e a largura do canal de saída não for suficiente para atender a demanda da rede, ocorrerá congestionamento [48].

Desta forma, faz-se necessário utilizar algoritmos de escalonamento para que os pacotes sejam enfileirados em um buffer de saída, evitando eventuais descartes de pacotes. Além disso, alguns algoritmos também permitem priorizar determinados tipos de tráfegos [48].

Assim, neste capítulo será apresentado o funcionamento das RSSF, enfatizando as redes de sensores corporais. A partir dos conceitos e funcionamento mostrados, será abordado o modelo de tráfego de uma rede de sensores corporal, relatando os parâmetros usados para a sua geração. Além disso, uma análise dos resultados e o impacto que as diferentes formas de geração de tráfego exercem será feito, tendo como base o modelo On/Off. Em seguida, os mecanismos de controle de acesso ao meio serão descritos, dando ênfase ao mecanismo *polling*, que foi o escolhido para ser implementado. O capítulo culminará com a análise dos dados gerados pelo simulador desenvolvido, o qual incorpora cinco modelos de tráfego para emular a geração de dados pelos nós da rede corporal, o mecanismo de controle de acesso ao meio *polling*, para organizar a disputa entre os nós para transmissão de pacotes, e o algoritmo de escalonamento FIFO desenvolvido em [115].

Para isso, foram definidos alguns cenários, possibilitando identificar os tempos de fila, de sistema e de serviço dos diversos tipos de tráfego simulados e também o descarte de pacotes, tanto nos nós sensores como no sorvedouro.

5.1 REDES DE SENSORES SEM FIOS (RSSF)

Um sensor é um dispositivo que emite sinais mensuráveis em resposta a uma mudança na condição física, sendo que estas medições devem ser reportadas ao interessado [77] e [120].

Um conjunto de sensores forma uma Rede de Sensores Sem Fio (RSSF), sendo que seus nós sensores podem monitorar, processar e comunicar vários fenômenos, a citar: monitoramento de dados em áreas hostis; variações de dados referentes à fisiologia humana, tais como pressão, temperatura, batimentos cardíacos; luminosidade; localização; entre outros [21], [98]. Contudo, geralmente os sensores possuem limitações no que se refere à capacidade de memória, poder computacional e energia. Além disso, o ponto fundamental em uma RSSF é o consumo de energia, pois as baterias são pequenas, possuindo pouca autonomia, [3] e [120].

Um nó sensor é composto basicamente por um transceptor, memória, processador, hardware para sensoriamento, sendo alimentados por uma bateria, conforme apresentado na Figura 5.1 [77] e [120].

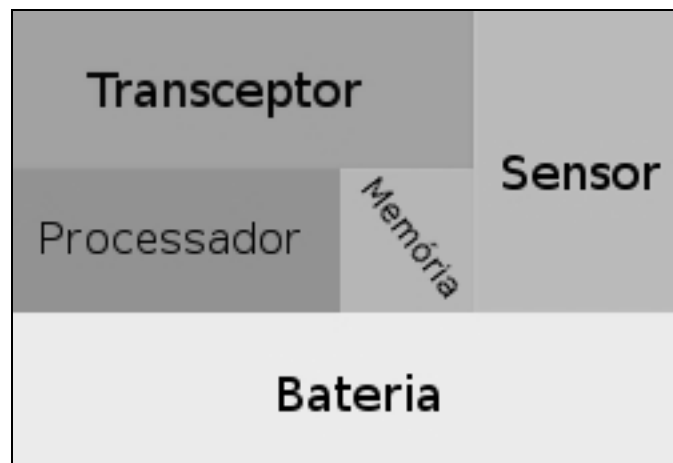


Figura 5.1: Componentes básicos de um nó sensor

Em função do tamanho reduzido, os nós sensores possuem limitações. Assim, a vida útil da rede se relaciona diretamente ao consumo de energia, sendo esta dependente da bateria utilizada. Deve ser levado em conta, também as diferenças entre as tecnologias de fabricação de baterias [77].

A RSSF é geralmente usada para monitoramento remoto em locais perigosos ou de difícil acesso, não existindo, portanto, uma infra-estrutura pré-determinada [21].

Há três funções que consomem energia em RSSF: o sensoriamento, processamento de informações e a comunicação entre os nós da rede. Dentre estas, a comunicação entre os nós é a que mais consome energia [54], [112].

Contudo, como a comunicação ocorre através de ondas de rádio, tendo o ar como meio de transmissão, se mais de um pacote iniciar a transmissão no mesmo instante, haverá colisão de pacotes. Se a colisão ocorrer, os pacotes envolvidos serão destruídos ou chegarão corrompidos ao destino. Isso implicará em retransmissão, causando desperdício de energia.

No caso de RSSF, onde os sensores estão distribuídos por grandes áreas geográficas, estas comunicações são complexas, pois além de tentar reduzir as colisões, cada nó deve descobrir uma rota para encaminhar os pacotes para o próximo nó.

Consoante ao nó centralizador, caso este faça a fusão de dados, que consiste no encaminhamento do resultado de um conjunto de amostras recebidas, haverá uma economia de energia.

Uma RSSF pode trabalhar no modo de monitoramento contínuo e no modo orientado a eventos. No primeiro, os sensores coletam dados a todo instante, sendo estes enviados ao nó centralizador. Como todas as medições feitas serão encaminhadas, os sensores tendem a ter uma vida útil reduzida. Já no modo orientado a evento, as informações são coletadas e só são enviadas se interessarem ao observador. Portanto este modo gera economia se comparado ao citado anteriormente, visto que há redução no número de informações transmitidas [54].

Mediante a diversidade de sensores e parâmetros envolvidos para o seu perfeito funcionamento, para a definição do nó centralizador, serão considerados os dados referentes à transmissão, consumo de energia, e ao tamanho e peso. Estas informações são apresentadas respectivamente nas Tabelas 5.1, 5.2 e 5.3.

Tabela 5.1: Parâmetros referentes à comunicação do nó centralizador.

Comunicação					
Nome	Alcance do sinal	Taxa de Transmissão	Frequência de banda	Potência de Radio Frequência	Consumo
SensiumTM Life Pebble TZ203082	5 m a 25 m	50 kbps	865 MHz a 928 MHz	-	-
Mica2	152,1 m em ambiente aberto	17 kbps	868 MHz a 916 MHz	-20 a +5 dBm	27 mA TX 10 mA RX <1 μ A sleep
Micaz	20 m a 30 m em ambiente fechado 75 m a 100 m em área aberta	250 kbps	2400 MHz a 2483.5 MHz	-24 dBm a 0 dBm	11 a 17,4 mA TX 19,7 mA RX <1 μ A sleep
Mica2Dot	-----	19.2 Kbps	868 MHz a 916 MHz	-20 - +5 dBm	27 mA TX 10 mA RX <1 μ A sleep

Fonte: [30], [121].

Conforme evidenciado na Tabela 5.1, os sensores de uma RSSF, em geral, possuem uma taxa de transmissão baixa se comparados com uma rede ad hoc convencional. Quanto ao alcance do sinal de cada nó, este é em média de 25 metros em ambiente fechado, possibilitando que o gateway se afaste um pouco dos nós. Assim, é possível ter uma aplicação em um ambiente como uma casa. Além disso, o gateway pode ser um computador com uma placa receptora ou um aparelho celular com Bluetooth.

Comparando o Mica2 com o MicaZ, a partir dos dados apresentados na Tabela 5.1, pode-se perceber que o MicaZ consome menos energia na transmissão, enquanto no Mica2 este fato ocorre na recepção. Quanto ao SensiumTM, não se obteve informações sobre o consumo do rádio. Já no que se refere ao modo sleep, que corresponde ao consumo quando o nó está em repouso (sem realizar processamento/transmissão), o Mica2 e MicaZ tem o mesmo gasto energético.

Na Tabela 5.2 constam dados referentes ao tempo de vida e tipo de bateria utilizada pelos sensores em análise.

Tabela 5.2: Configuração do sensor quanto a Energia.

Energia		
Nome	Tipo de bateria	Tempo de operação padrão
SensiumTM Life Pebble TZ203082	Zinc Air IEC PR44 (675) Ou LR44/A76 Alcalina	120 hrs de uso contínuo
Mica2	2 pilhas AA	aprox. 450 recepções e 18 transmissões
MicaZ	2 pilhas AA	aprox. 236 recepções e 45 transmissões
Mica2Dot	Coin Cell de 3 V	-----

Fonte: [30], [121].

Pode-se observar, através da Tabela 5.2, que os sensores sem fio em análise possuem uma fonte energética limitada e de baixa capacidade. No entanto, as baterias dos sensores MicaZ e Mica2 são facilmente encontradas. Como para estes sensores não se obteve informações consoante ao tempo de operação padrão, estes foram estimados baseando-se nos seguintes fatos:

- a pilha AA possui 2500 mAh [8];
- os valores de consumo de transmissão e recepção foram apresentados na Tabela 5.1;
- em [98] é mencionado que a taxa de transmissão corresponde a 10% do tempo. Então se conclui que 90% ele passa dormindo;
- como os sensores MicaZ e Mica2 podem realizar fusão e centralização dos dados, o seu rádio estará no modo de recepção a maior parte do tempo.

Com base nestes fatos, os cálculos foram realizados considerando que o consumo ocorrerá 90% do tempo para recepção de dados e 10% para transmissões.

Consoante as dimensões dos sensores, estas se encontram Tabela 5.3.

Tabela 5.3: Configuração do sensor quanto à dimensão e peso.

Medidas		
Nome	Dimensões	Peso
SensiumTM Life Pebble TZ	50 mm de diâmetro x 13 milímetros de espessura	-
Mica2	58 mm de comprimento x 32 mm de largura x 7mm de altura (sem a bateria)	18 gramas
MicaZ	58 mm de comprimento x 32 mm de largura x 7mm de altura (sem a bateria)	18 gramas
Mica2Dot	25 mm de diâmetro por 6 mm de altura, (sem a bateria)	3 gramas

Fonte: [30], [121].

Pode-se notar, pela Tabela 5.3, que não há diferenças expressivas quanto às dimensões e pesos dos sensores que podem ser usados como centralizadores, sendo que o Mica2Dot é menor que os demais.

No entanto, dependendo da aplicação, em [30] é relatado que o sensor MicaZ pode ser usado como estação base, recebendo assim os dados dos outros sensores que compõe a RSSF. Ao receber as informações, estas podem ser enviadas para uma estação com conexão com a Internet.

Isso é possível graças ao seu rádio com uma capacidade maior de transmissão, além da facilidade na substituição das pilhas.

5.2 REDE DE SENSORES CORPORAIS

Uma RSSF composta por sensores biológicos que visam monitorar sinais fisiológicos de seres humanos é chamada de Rede de Sensores Corporais (RSC). Estas redes, compostas por vários nós sensores com capacidade limitada de processamento, comunicação e energia, monitoram diversas atividades do corpo humano, facilitando o acompanhamento médico de pacientes que requerem uma maior atenção.

Pacientes monitorados através de RSC podem desempenhar suas atividades normalmente, pois o monitoramento é remoto e o médico recebe as informações do estado de saúde em tempo real. Isso possibilita redução nos gastos com internações, minimizando a ocupação de leitos e o risco de infecções hospitalares [12]. Com isso, percebem-se os benefícios das RSC, sendo um exemplo demonstrado na Figura 5.2.

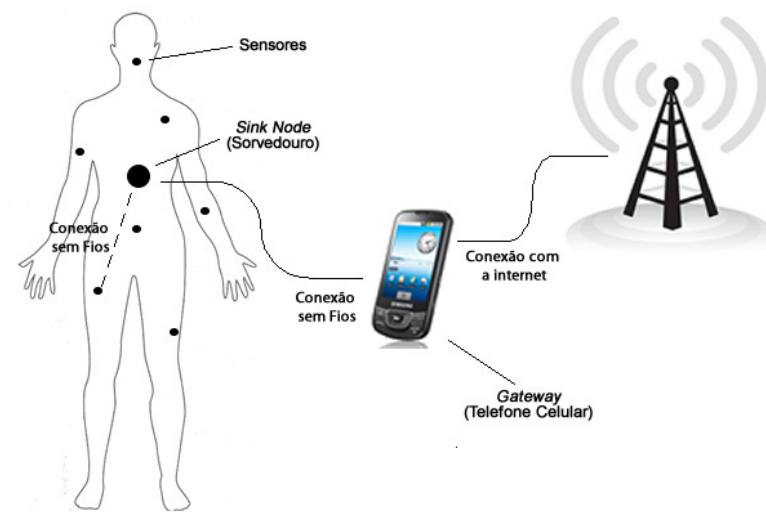


Figura 5.2: Redes de Sensores para o Corpo Humano

Conforme pode ser visualizado na Figura 5.2, vários sensores estão dispostos no corpo humano, sendo que estes podem estar localizados interna ou externamente ao corpo. Os nós

realizam a coleta de informações e encaminham ao sorvedouro, que possui maior capacidade de armazenamento e processamento. Por sua vez, o sorvedouro faz a conexão entre os sensores e as redes externas.

Uma vez que os sensores transmitem suas informações para o nó sorvedouro, e de acordo com [71] e [136] a estrutura da rede é uma estrela, o nó centralizador foi inserido no meio dos sensores. Além disso, é no nó sorvedouro que está localizado o mecanismo de controle de acesso ao meio.

Percebe-se ainda pela Figura 5.2 que os sensores são pequenos se comparados ao sorvedouro. Além disso, eles devem ser leves para não gerar desconforto em pacientes que usam o dispositivo por longos períodos de tempo [129]. Porém, em função do tamanho, suas limitações são ainda maiores. Outrossim, o tamanho e o peso do sensor são determinados principalmente pelas dimensões da bateria [60] e [98]. Assim, os requisitos para aumentar a vida útil da bateria são inversamente proporcionais aos requisitos do sensor. Esta afirmação corrobora com o que foi mencionado em [3] referente ao consumo de energia das RSSF. Assim, como uma RSC é uma RSSF, o mesmo fato ocorre. No entanto, em uma RSC, dependendo da localização do sensor, a bateria pode não ser um recurso renovável ou de substituição possível [120].

Tabela 5.4: Comparação entre as configurações dos sensores Mica2Dot, o Mica2 e o MicaZ.

Modelo	MICA2DOT	MICA2	MICAz
Bateria	3V coin cell	AA × 2	AA × 2
Tamanho (mm)	25 × 6	58 × 32 × 7	58 × 32 × 7
Peso (g)	3	18	18
Alcance (m)	150	150 a 300	75 a 100
Taxa de transmissão (Kbps)	38.4	38.4	250
Memória flash programável	128 KB	128 KB	128 KB
Memória flash serial	512 KB	512 KB	512 KB
EEPROM	4 KB	4 KB	4 KB
RF (MHz)	315/433/868/916	315/433/868/916	2400

Analisando a Tabela 5.4 percebe-se que no que se refere aos tipos de memórias, não há diferença entre os sensores. No que tange a frequência do rádio, o Mica2Dot e o Mica2 trabalham na mesma frequência.

Contudo, as dimensões do sensor Mica2Dot são bem inferiores aos demais. Isso faz com que o mesmo atenda ao requisito de tamanho e leveza, desejáveis para aplicações em RSC. Porém sua taxa de transmissão é seis vezes menor do que o MicaZ e igual a do Mica2.

Deste modo, em função de ter uma taxa de transmissão bem elevada se comparado ao Mica2Dot e Mica2, o sensor MicaZ pode ser usado como sorvedouro.

Consoante a composição das RSC, em [129] é relatado que estas redes são formadas por sensores heterogêneos, sendo estes específicos para atender uma determinada necessidade, como o monitoramento de temperatura, por exemplo. Estes sensores fazem o monitoramento das atividades biofísicas, coletando, processando e reportando esses dados para uma estação onde é realizado o processamento, análise e gerados relatórios.

No que tange ao monitoramento, de acordo com o que foi relatado, as RSSF podem operar no modo monitoramento contínuo e no modo orientado a eventos. Em RSC o ideal é que os sensores façam o monitoramento contínuo para detectar previamente algum problema, antes que o mesmo se alastre. Contudo, nem todas as medições precisam ser enviadas. Por exemplo, sendo a temperatura corporal considerada normal próxima a 36º C, se não houver variações que comprometam o estado de saúde do paciente, a maioria das medições não precisam ser enviadas.

Entretanto, se o sorvedouro passar um longo intervalo de tempo sem receber dados de um sensor específico, pode deduzir que o sensor está inativo. Uma forma de evitar este problema é o envio de mensagens de controle, demonstrando que o sensor está ativo e o paciente sem febre. Estas mensagens poderiam ser enviadas a cada trinta (30) medições realizadas, por exemplo.

5.3 MODELOS DE FONTES PARA RSC

Conforme relatado no Capítulo 3, para analisar o comportamento de um sistema de redes, seus usuários e aplicações, são utilizados modelos de tráfego. Isso não é diferente nas RSC.

Além disso, como descrito no Capítulo 3, uma fonte de tráfego consiste em um dispositivo que permite a geração de tráfego [88], (p. 07). Contudo, o modelo de fonte deve ser fidedigno às características do ambiente em análise.

Na análise feita referente aos modelos de tráfego para redes convergentes, dos modelos existentes, o mais simples é o modelo On/Off [16]. Embora este modelo seja utilizado para redes convergentes, o mesmo pode ser empregado em RSC. Afinal, quando os sensores não possuem pacotes para transmitir podem desligar o rádio [18], [51], sendo uma analogia ao intervalo Off. Assim, há períodos onde há transmissão de dados, que seriam os intervalos On, e períodos onde os nós estão em repouso, não havendo atividade no canal de comunicação da rede, representando os períodos Off. Deste modo, no intervalo Off os sensores podem dormir ou realizar outras tarefas, como coletar e processar os dados.

No entanto, para utilizar o modelo On/Off, os parâmetros para a geração dos tráfegos devem ser adequados à realidade das RSC.

Contudo, antes que este modelo seja empregado, faz-se necessário analisar trabalhos similares, para se assegurar que o modelo On/Off reflete a realidade do tráfego em RSC.

Em [108] são utilizados dois Modelos Markovianos. Um com modulação poissoniana e outro determinística, ambos para modelar a geração de dados de um sensor de eletroencefalograma. O modelo poissoniano foi usado para modelar o sinal do eletroencefalograma e o determinístico foi utilizado para a montagem do número de bits necessários para codificar cada valor do sinal do eletroencefalograma (e indiretamente, a energia usada para transmitir o sinal) [108]. Os modelos de tráfego Markovianos têm sido frequentemente utilizados para caracterizar o tráfego em redes de comunicação, o que foi comentado no Capítulo 3.

O modelo poissoniano para a geração do tráfego também é usado em [124] e [119]. Em [119], os pacotes são armazenados em buffer enquanto não são transmitidos e a ordem de envio dos pacotes nos nós é FIFO.

O modelo auto-regressivo é apresentado em [89], onde os resultados mostram que a geração de pacotes segue uma distribuição Gaussiana aleatória. O estudo é realizado em um nó que monitora a temperatura corporal e sinais do eletrocardiograma, com uma frequência de medidas de 360 medidas por segundo.

Em [74] é reforçada a necessidade de um correto modelo de tráfego para avaliar o desempenho de todo o sistema. Além disso, afirmam que os nós das redes de sensores sem fios podem operar em modo contínuo, com coletas periódicas de dados e em modo orientado a evento, onde os dados são gerados quando um evento esperado ocorre no ambiente. Deste modo, afirmam que em uma RSSF podem coexistir nós operando nos dois modos. Assim, os nós de monitoramento contínuo podem ser modelados de acordo com o modelo poissoniano ou *Constant Bit Rate* (CBR). Já os orientados a evento podem ser modelados como fontes auto-similares, pois a princípio não se conhece o comportamento do fenômeno monitorado, uma vez que pode haver períodos onde os nós não estão gerando dados e períodos em que os dados podem ser gerados em rajadas devido a um estímulo esperado. Diante disto em seu trabalho são modeladas fontes On/Off auto-similares.

Desde o surgimento do modelo auto-similar para representar o tráfego de redes, há desafios para a sua modelagem [74]. Deste modo, devido à grande complexidade e dificuldade em se calcular os parâmetros para a modelagem do tráfego utilizando *Fractional Gaussian Noise* (FGN), *Fractional Brownian Motion* (FBM), *Fractional AutoRegressive Integrated Moving Average* (FARIMA) entre outros, a escolha destes métodos para simulação das fontes não é comum. Assim, justificam a escolha do modelo On/Off tradicional, por ser um modelo prático, simples, fácil de controlar, além de representar com boa aproximação o comportamento auto-similar. Vale ressaltar que este fato já foi mencionado na revisão de literatura apresentada no Capítulo 3.

Não existe ainda um consenso, na literatura, para os modelos de fontes para RSC. Afinal, pode haver sensores monitorando fenômenos com pouca variação no tempo e outros, na mesma

rede, monitorando outros fenômenos com uma taxa de variação maior. Também são influenciados pelo modo de operação dos nós que podem operar em modo contínuo, com medições periódicas, e no modo orientado a evento onde há uma filtragem dos dados que serão aproveitados.

Neste trabalho, foi escolhido o modelo On/Off exponencial devido à sua simplicidade e pelo fato de gerar fluxo de tráfego com boa aproximação ao modelo auto-similar, como ocorre em redes convergentes.

5.3.1 Fontes Propostas para RSC

Os parâmetros para geração de fontes On/Off utilizados neste trabalho são mostrados na Tabela 5.5.

Tabela 5.5: Parâmetros para a geração das fontes

Tamanho do Pacote	904 bits
Taxa de pico	39321,6 bps
Intervalo On	0,022988 s
Intervalo Off	0,20690 s

Para a obtenção do tamanho do pacote, fez-se uma média dos tamanhos de pacotes utilizados em [22], [49], [71], [92], [95], [96], [116] e [126]. É importante mencionar que todos os artigos considerados se referem a RSC e que não há muita discrepância entre os valores dos tamanhos de pacotes apresentados, exceto o artigo [49], que adota o valor de 96 bits.

No que tange a taxa de pico, como o nó Mica2Dot é mencionado na maioria das literaturas pesquisadas, o valor apresentado consta na Tabela 5.4 e corresponde a taxa de transmissão, que para este sensor é 38,4 Kbps. Então apenas foi realizada a conversão para a unidade bits por segundo (bps). Já o tempo de On, informado na Tabela 5.5, corresponde ao tamanho do pacote dividido pela taxa de pico. Quanto ao tempo de Off, este foi obtido através de regra de três simples, considerando que os sensores ficam 90% no estado Off [98], [112].

Desta forma, tendo definido o modelo On/Off para a geração dos tráfegos, bem como os parâmetros, cinco fontes foram propostas, sendo que a descrição e análise são relatadas a seguir.

Vale salientar que para fins estatísticos, em todas as análises realizadas, assim como nas fontes para redes convergentes, o percentual de vinte por cento (20%) dos pacotes iniciais são desconsiderados. Isso é realizado para que o sistema possa entrar em equilíbrio. Assim, nas simulações onde 10000 pacotes são gerados, 8000 são considerados.

5.3.1.1 Fonte On/Off Constante

Para simular o comportamento de uma RSC, o simulador Fonte On/Off Constante, descrito no item 3.5.1, foi adaptado.

Conforme relatado, este simulador gera um pacote a cada intervalo On, sendo que este pacote tem tamanho fixo. Desta forma os intervalos On têm a mesma duração, pois são obtidos através da divisão do tamanho do pacote pela taxa de pico. Já para o intervalo Off é definido um valor médio, e com base neste valor é gerada a duração do intervalo, usando a distribuição exponencial negativa.

A Fonte On/Off Constante opera no modo de monitoramento contínuo, no qual os sensores estão sempre coletando informações e enviando ao sorvedouro ou servidor. Assim, os sensores tendem a ter uma vida menor, já que todas as medições realizadas são enviadas [112].

Através da inserção dos parâmetros apresentados na Tabela 5.5 no simulador, os quais emulam o comportamento de uma RSC com monitoramento contínuo, ensaios foram feitos com 10000, 20000 e 30000 pacotes no sistema. Para maior confiabilidade, cada simulação foi executada três vezes, sendo a média das três simulações apresentadas na Tabela 5.6.

Tabela 5.6: Resultados das simulações com a Fonte On/Off Constante

Fonte Constante	10000	20000	30000
Quant. Intervalos On	8000	16000	24000
Totalizador de On (s)	183,9260	367,8117	551,7023
Quant. Intervalos Off	8000	16000	24000
Totalizador de Off (s)	1692,9462	3370,1124	5071,5677
Média de On (s)	0,0230	0,0230	0,0230
Média de Off (s)	0,2116	0,2106	0,2113

Analisando a Tabela 5.6 pode-se constatar que os resultados são condizentes com o esperado. Afinal, a quantidade de intervalos On e Off é a mesma. Além disso, como os primeiros vinte por cento (20%) dos pacotes são desconsiderados e após um intervalo de On há um de Off, os resultados estão corretos.

Consoante ao tempo de On, uma vez que os pacotes têm tamanhos fixos e só é gerado um pacote a cada intervalo ativo, se for dividido o valor do tamanho do pacote pela taxa de pico, ambos informados na Tabela 5.5, obtém-se a média de On apresentada na Tabela 5.6. Além disso, se for dividido o totalizador de On pela quantidade de intervalos ativos, a mesma média de On é obtida, confirmando que o programa está funcionando corretamente. Este último procedimento também é válido para o Intervalo de Off. Porém a média de Off não corresponde exatamente ao valor informado na Tabela 5.5. Isso se justifica pelo fato que seu valor é gerado com distribuição exponencial. Contudo, como a literatura menciona que os sensores passam aproximadamente 90% em silêncio, o valor obtido é próximo, sendo que neste caso corresponde a 90,17% nas simulações com 10000 pacotes.

Visando economia de energia, quatro novas fontes, baseadas na Fonte On/Off Constante, foram desenvolvidas. Estas possuem características que representam o comportamento dos sensores no modo de monitoramento orientado a evento, onde os sensores enviam somente informações relevantes ao observador interessado no evento. Estas propostas são apresentadas e analisadas a seguir.

5.3.1.2 Fonte On/Off Limiar

A fonte On/Off Limiar faz o monitoramento dos dados e envia somente as medidas que forem acima de um determinado valor. Por exemplo, se for definido como limite mínimo para o envio de pacotes o valor 36, caso a temperatura corporal de uma pessoa se mantenha a 36°C, então poucas transmissões serão realizadas, sendo esta restrita aos pacotes que apresentarem valores acima de 36°C.

Baseado nesta premissa foi criada uma função que gera valores aleatórios com distribuição exponencial para representar a medida realizada pelo sensor. Para selecionar os pacotes que devem ser transmitidos ou descartados, o valor gerado é comparado com um parâmetro fornecido durante a configuração da fonte.

Esta fonte, desenvolvida em [104], é aconselhável para monitorar sinais fisiológicos, onde qualquer alteração mínima represente risco à saúde, sendo estes pacotes transmitidos. Assim, para testar o sistema foi adotado o valor mínimo de 0,66, pois se considerou medições referentes à temperatura, pressão e batimentos cardíacos, e que apenas a temperatura não tem muita variação. Desta forma, as medições cuja informação for acima deste valor serão transmitidas.

Os resultados obtidos com as simulações da Fonte On/Off Limiar constam na Tabela 5.7.

Tabela 5.7: Resultados das simulações com a Fonte On/Off Limiar

Fonte Limiar	10000	20000	30000
Quant. Intervalos On	2972	5892	8960,6667
Totalizador de On (s)	68,3346	135,4525	206,0165
Quant. Intervalos Off	8000	16000	24000
Totalizador de Off (s)	1680,1430	3346,3137	5029,3314
Média de On (s)	0,0230	0,0230	0,0230
Média de Off (s)	0,2100	0,2091	0,2096
Pacotes menores que o Limiar	5028	10108	15039
Pacotes maiores que o Limiar	2972	5892	8961
Média das informações	0,6661	0,6687	0,6663
Totalizador de Off dos pacotes não transmitidos (s)	115,5813	232,3994	345,7358
Tempo Total de Off (s)	1795,7243	3578,7132	5375,0672
Média de Off (s)	0,2245	0,2237	0,2240

Através da Tabela 5.7 constata-se que 20% dos pacotes iniciais não foram considerados para fins estatísticos, o que pode ser comprovado pela quantidade de intervalos Off. Também nota-se que a média de On converge para o valor definido na Tabela 5.5, assim como a quantidade de pacotes maior do que o limiar é equivalente a quantidade de intervalos On.

Além disso, somando a quantidade de pacotes maiores do que o limiar com o número de pacotes menores do que o limiar, obtém-se a quantidade de pacotes gerados, ou seja, 8000 pacotes nas simulações com 10000 pacotes no sistema.

Também se percebe que o totalizador de On está correto, pois multiplicando o tamanho de cada intervalo ativo pela quantidade de pacotes transmitidos chega-se ao valor de 68,3346 s.

Outro fator relevante é a convergência do valor definido como limiar, sendo este de 0,66.

Analisando o totalizador de Off, percebe-se que os valores obtidos são muito próximos do que foi apresentado na Tabela 5.6. No entanto, quando um pacote não é transmitido, o tempo do intervalo On deve ser contabilizado como sendo intervalo Off. Por isso é apresentado o totalizador de Off dos pacotes não transmitido. Isso resulta no aumento da média do intervalo Off, correspondendo a aproximadamente 6% dos totalizadores de Off apresentados na Tabela 5.6, onde todos os pacotes foram transmitidos.

5.3.1.3 Fonte On/Off Limiar Controlado

Este modelo de fonte baseia-se na Fonte On/Off Limiar e parte do pressuposto que durante um tempo muito longo os nós não transmitam nenhum pacote, pois nenhuma leitura realizada atende ao critério especificado para a transmissão, ou seja, todos os valores medidos são inferiores ao definido. Desta forma, para que sempre haja um histórico atualizado sobre o sinal monitorado, foi implementada uma função que conta quantos pacotes foram descartados, e quando este contador atinge determinado valor, o próximo pacote é enviado independente de sua medição satisfazer ou não o critério estabelecido.

Assim, esta fonte além de informar o status do sensor, permite que as informações no servidor estejam sempre atualizadas. Afinal, se transcorrer muito tempo sem o envio de informações, pode-se imaginar que o sensor não esteja mais ativo.

Visando verificar o funcionamento desta fonte, simulações foram realizadas considerando o valor limite de envio do pacote 0,66 e a cada dez pacotes não transmitidos, um pacote de controle é enviado.

Os resultados referentes à Fonte On/Off Limiar Controlado são apresentados na Tabela 5.8.

Tabela 5.8: Resultados das simulações com a Fonte On/Off Limiar Controlado

Fonte Limiar Controlado	10000	20000	30000
Quant. Intervalos On	3055	5939	9015
Totalizador de On (s)	70,2272	136,5410	207,2736
Quant. Intervalos Off	8000	16000	24000
Totalizador de Off (s)	1688,2246	3348,4986	5041,9559
Média de On (s)	0,0230	0,0230	0,0230
Média de Off (s)	0,2110	0,2093	0,2101
Pacotes menores que o limiar	4945	10061	14985
Pacotes maiores que o limiar	3043	5904	8968
Média das informações	0,6758	0,6642	0,6700
Totalizador de Off dos pacotes não transmitidos (s)	113,6887	231,3110	344,4796
Tempo Total de Off (s)	1801,9133	3579,8096	5386,4355
Média de Off (s)	0,2252	0,2237	0,2244
Pacotes de Controle	12	35	48

Mediante os resultados apresentados na Tabela 5.8 pode-se comprovar que os dados convergem para os parâmetros da Tabela 5.5, sendo que o mesmo ocorre com a média da informação definida. Com isso pode-se dizer que o simulador está funcionando corretamente. Outra prova disso é que a quantidade de intervalos Off corresponde exatamente ao número de pacotes considerados para fins estatísticos. Além disso, se for somado o número de pacotes maiores e menores que o limiar, mais os pacotes de controle, se obtêm justamente o número de pacotes considerados para fins estatísticos.

Nota-se, também, mediante os dados apresentados na Tabela 5.8, que o totalizador de Off não tem um acréscimo considerável em função de intervalos ativos inexistente, sendo equivalente a aproximadamente 6%. Assim, constata-se que o acréscimo do pacote de controle não gera impacto no sistema no que se refere ao intervalo Off, quando comparado a Fonte On/Off Limiar.

No entanto, com a inserção do pacote de controle, o número de pacotes transmitidos aumenta aproximadamente 3%. Isso pode ser comprovado, pois quando este parâmetro não existia, o número de pacotes transmitidos foi 2972 nas simulações com 10000 pacotes no sistema (Tabela 5.7). Introduzindo esta informação, a quantidade de pacotes transmitidos é de 3055, nas simulações com 10000 pacotes (Tabela 5.8), o que corresponde a 83 pacotes a mais. Este valor é ínfimo se analisado os benefícios que o parâmetro proporciona. Afinal, o sorvedouro não precisará verificar quais os nós estão ativos. Assim, não há necessidade da utilização do canal de controle ou de multiplexar o canal para verificar o status dos sensores.

5.3.1.4 Fonte On/Off Fora-Faixa

Para simular os sensores que somente enviam medidas que estejam fora de determinado intervalo, deve-se informar ao programa dois parâmetros para que este intervalo seja calculado. Estes parâmetros são o valor médio e o percentual de variação sobre este valor. Por exemplo, em um sensor responsável pelo monitoramento da pressão arterial, deseja-se que somente as medidas que representem riscos para o paciente sejam enviadas. Considerando que o normal para determinado paciente seja de 100 batimentos por minuto e que estes batimentos possam variar entre 80 e 120 por minuto, deve-se então informar ao programa como valor médio o valor 100 e o percentual de variação 20%. Desta forma, somente serão enviados os pacotes que apresentarem uma leitura de batimentos cardíacos menores que 80 ou maiores que 120.

A proposta da Fonte On/Off Fora-Faixa é justamente atender aplicações deste gênero, ou seja, transmitir somente pacotes que apresentem variação acima ou abaixo da faixa definida. Assim, esta fonte é indicada para representar o monitoramento de sinais, onde nem todas as alterações representam risco à saúde, como o caso mencionado da frequência cardíaca.

Para testar o sistema e confrontar os resultados com as fontes descritas anteriormente, o valor médio usado foi 0,66 e o percentual de variação foi 66%. Assim, é necessário identificar quais as informações que deverão ser enviadas. Para isso, basta calcular o percentual de variação do valor médio informado, e para a faixa superior adicionar o valor médio, ou seja, calcular 66% de 0,66 e ao valor obtido, somar 0,66. Então, somente os pacotes que apresentarem medidas acima de 1,1 e abaixo de 0,22 serão transmitidos.

Os resultados das simulações com a Fonte On/Off Fora-Faixa constam na Tabela 5.9.

Tabela 5.9: Resultados das simulações com a Fonte On/Off Fora-Faixa

Fonte Fora-Faixa	10000	20000	30000
Quant. Intervalos On	3793	7508	11268
Totalizador de On (s)	87,2072	172,6201	259,0793
Quant. Intervalos Off	8000	16000	24000
Totalizador de Off (s)	1673,8483	3331,2813	5021,3504
Média de On (s)	0,0230	0,0230	0,0230
Média de Off (s)	0,2092	0,2082	0,2092
Pacotes dentro da faixa	4207	8492	12732
Pacotes fora da faixa	3793	7508	11268
Média das informações	0,6693	0,6669	0,6653
Totalizador de pacotes dentro da faixa (s)	96,7087	195,2319	292,7063
Tempo Total de Off (s)	1770,557	3526,5132	5314,0567
Média de Off (s)	0,2213	0,2204	0,2214
Informações acima da faixa	1570	3076	4595
Informações abaixo da faixa	2223	4432	6673

Através da Tabela 5.9 pode-se comprovar que a média das informações está convergindo para o valor informado, ou seja, 0,66. Também se pode demonstrar que o simulador está funcionando corretamente através da soma da quantidade de pacotes dentro e fora da faixa que totalizam 8000 pacotes. Estes 8000 pacotes também foram obtidos para a quantidade de intervalos Off.

Como a forma de geração do intervalo Off não é alterada em nenhuma fonte, ou seja, todos os intervalos são gerados com distribuição exponencial com média de 0,20690 s, definida na Tabela 5.5, percebe-se que os valores obtidos estão muito próximos ao idealizado. Isso pode ser

comprovado, pois como se observa na Tabela 5.9, o tempo médio de Off é 0,2213 s. Isso representa uma diferença de 0,0144 s o que é considerada pequena.

Consoante ao tempo de On, dividindo o totalizador de On pela quantidade de intervalos On, se obtém o valor de 0,0229916 s, o que é praticamente o valor definido na Tabela 5.5 que é 0,022988 s. Além disso, a média do intervalo On obtida na Tabela 5.9 é basicamente a mesma. Como os pacotes dentro da faixa não devem ser transmitidos, o que corresponde a 96,7087 s, dividindo este valor pela quantidade de pacotes dentro da faixa, tem-se o resultado de 0,022987 s, o que comprova que o tempo de On converge também neste sentido.

Além disso, usando esta fonte, os sensores permanecem 95,05% do tempo com o rádio desligado. Este valor é obtido fazendo uma regra de três simples, na qual os totalizadores de On e Off somados correspondem a 100% do tempo.

Outra informação que comprova que o simulador está funcionando corretamente é que somando as informações acima e abaixo da faixa, as quais correspondem aos pacotes que devem ser transmitidos, se obtém exatamente o valor 3793 pacotes. Este resultado é justamente o que consta em pacotes fora da faixa, sendo também equivalente a quantidade de intervalos On.

5.3.1.5 Fonte On/Off Fora-Faixa Controlada

Esta fonte é muito similar a Fonte On/Off Fora-Faixa. A única alteração é que nela o parâmetro controle é inserido. Assim, nesta fonte só são transmitidos pacotes com medidas acima ou abaixo do percentual definido, ou se um determinado número de pacotes foi descartado desde a última transmissão, um pacote será enviado, independente do valor da informação.

Os resultados das simulações com esta fonte são apresentados na Tabela 5.10. Para as simulações foram adotados os seguintes valores como parâmetros: valor médio 0,66, percentual de variação de 66% e um pacote de controle enviado a cada dez pacotes não transmitidos.

Tabela 5.10: Resultados das simulações com a Fonte On/Off Fora-Faixa Controlada

Fonte Fora-Faixa Controlada	10000	20000	30000
Quant. Intervalos On	3763	7555	11297
Totalizador de On (s)	86,5099	173,6855	259,7380
Quant. Intervalos Off	8000	16000	24000
Totalizador de Off (s)	1690,7224	3321,0195	5028,8932
Média de On (s)	0,0230	0,0230	0,0230
Média de Off (s)	0,2113	0,2076	0,2095
Pacotes dentro da faixa	4237	8445	12703
Pacotes fora da faixa	3758	7548	11288
Média das informações	0,6722	0,6663	0,6683
Totalizador de pacotes dentro da faixa (s)	97,4060	194,1664	292,0475
Tempo Total de Off (s)	1788,1283	3515,1859	5320,9407
Média de Off (s)	0,2235	0,2197	0,2217
Informações acima da faixa	1551	3087	4649
Informações abaixo da faixa	2207	4461	6640
Mensagens de controle	5	7	9

Observando a Tabela 5.10, comprova-se que a média das informações é condizente com o que foi definido, ou seja, 0,66. Isto prova que a geração de números aleatórios está convergindo para o valor definido. A quantidade de intervalos On é equivalente a quantidade de pacotes que foram transmitidos, que na Tabela 5.10 corresponde ao número de pacotes fora da faixa mais as mensagens de controle. Outrossim, somando a quantidade de pacotes fora e dentro da faixa com as mensagens de controle, se obtém 8000 pacotes, que corresponde ao número de pacotes gerados, bem como a quantidade de intervalos Off.

A média de On apresentada na Tabela 5.10 converge para o número definido na Tabela 5.5, sendo que se for dividido o totalizador de On pela quantidade de intervalos On, o mesmo valor será obtido. Isso é mais uma comprovação de que o simulador está funcionando conforme especificado.

Além disso, somando as informações acima e abaixo da faixa, o valor de 3758 pacotes é obtido, convergindo para os pacotes fora da faixa e que devem ser transmitidos.

Quanto a média de Off, observa-se que o valor obtido na Tabela 5.10 é similar ao obtido na Tabela 5.9, que corresponde a mesma fonte sem a inserção do parâmetro controle. O mesmo acontece com o tempo total de Off. Efetuando uma regra de três, pode-se perceber que os sensores permanecem em silêncio 95,38% do tempo, o que é um valor intermediário ao que foi definido em [98] e [112].

5.3.2 Análise das Fontes Propostas

A partir das fontes orientadas a evento propostas, pode-se constatar que as mesmas reduzem o número de transmissões, cumprindo com seu propósito. Além disso, as fontes que usam o parâmetro controle possibilitam a identificação do status dos sensores, sem para isso utilizar o canal de controle, ou mesmo multiplexar o canal. Vale ressaltar que o acréscimo de transmissões de pacotes de controle não é algo expressivo, tendo em vista as vantagens providas.

Visando comprovar a redução nas transmissões usando as fontes orientadas a eventos, bem como que o parâmetro controle não ocasiona um aumento expressivo de pacotes na rede, um gráfico comparando a quantidade de intervalos ativos das fontes propostas é apresentado na Figura 5.3.

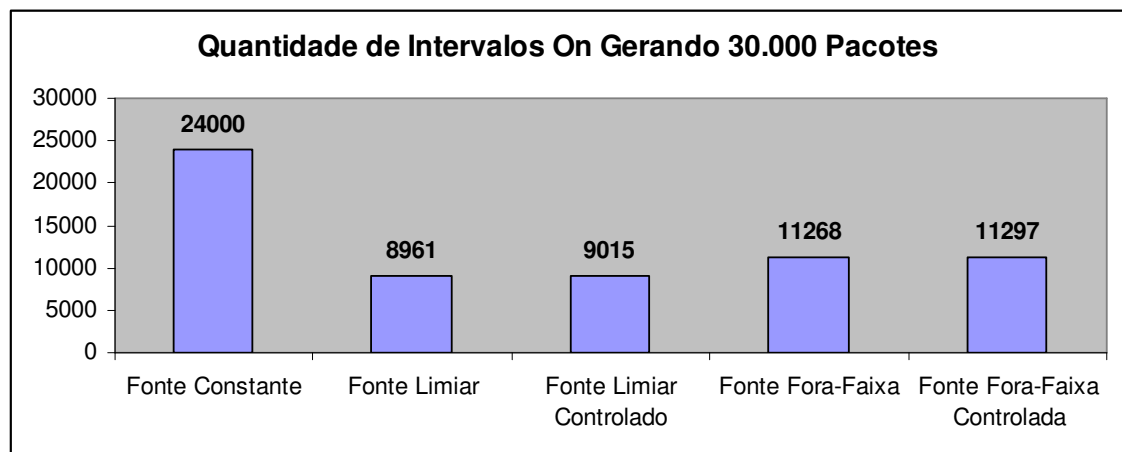


Figura 5.3: Comparativo entre a quantidade de intervalos On

Analisando a Figura 5.3 constata-se que a redução usando as fontes orientadas a evento é expressiva, se comparado a fonte Constante. Este decréscimo é de aproximadamente 53% no pior caso, correspondendo a Fonte Fora-Faixa Controlada. Com a Fonte Limiar a redução nas transmissões corresponde a aproximadamente 63%.

Também é possível perceber que com a introdução do parâmetro controle não há um consumo de energia considerável, tendo em vista que na fonte Limiar Controlado houve um aumento de 54 pacotes se comparado a mesma fonte sem o parâmetro controle. Já na Fonte Fora-Faixa Controlada este acréscimo foi de 29 pacotes. Desta forma, na pior situação, que corresponde a Fonte Limiar Controlado, este aumento não chega a um por cento (1%), sendo praticamente desprezível.

Com base nos números apresentados constata-se que o método de monitoramento contínuo pode ser usado de modo mais eficiente. Afinal, o paciente continuará sendo monitorado a todo o momento, sendo que o tráfego na rede se restringirá as transmissões das informações necessárias, ou seja, apenas em casos de alertas.

Para ficar mais evidente a economia de energia proporcionada pelas fontes orientadas a evento, um comparativo referente ao tempo total dos intervalos ativos é mostrado na Figura 5.4.

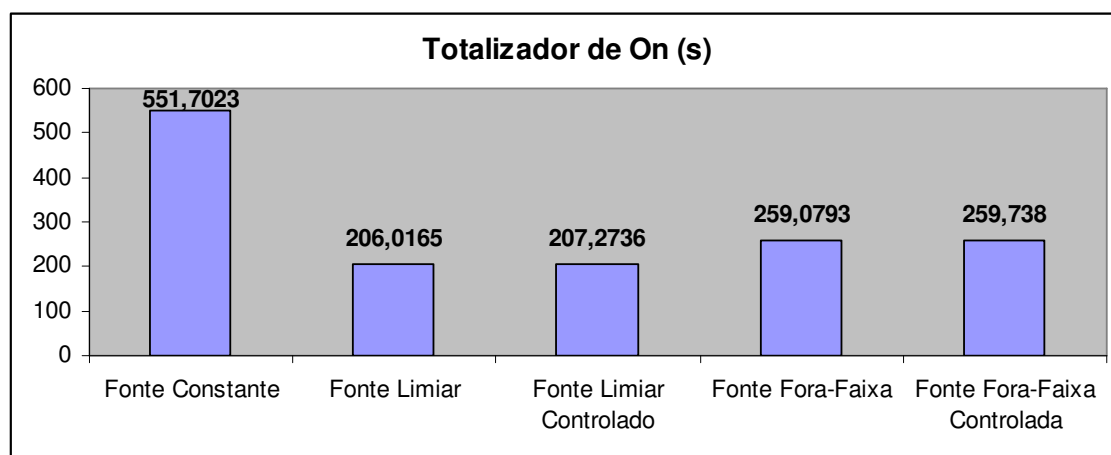


Figura 5.4: Análise do tempo total dos intervalos On

Analisando a Figura 5.4 pode-se perceber que os valores apresentados estão convergindo para o intervalo de On mencionado na Tabela 5.5, ou seja, 0,022988 s. Isso pode ser comprovado

dividindo o totalizador de On da Fonte Constante, mostrado na Figura 5.4, pela quantidade de intervalos On da Fonte Constante, apresentada na Figura 5.3. Isso vale para as demais fontes.

Além disso, comprova-se que a Fonte Constante deixa o sorvedouro mais tempo ocupado, tendo em vista que todos os sensoriamentos realizados são transmitidos.

De modo a visualizar a fonte que permite com que o sorvedouro fique mais tempo ocioso, a Figura 5.5 compara o tempo total dos intervalos Off, expressos em segundos.

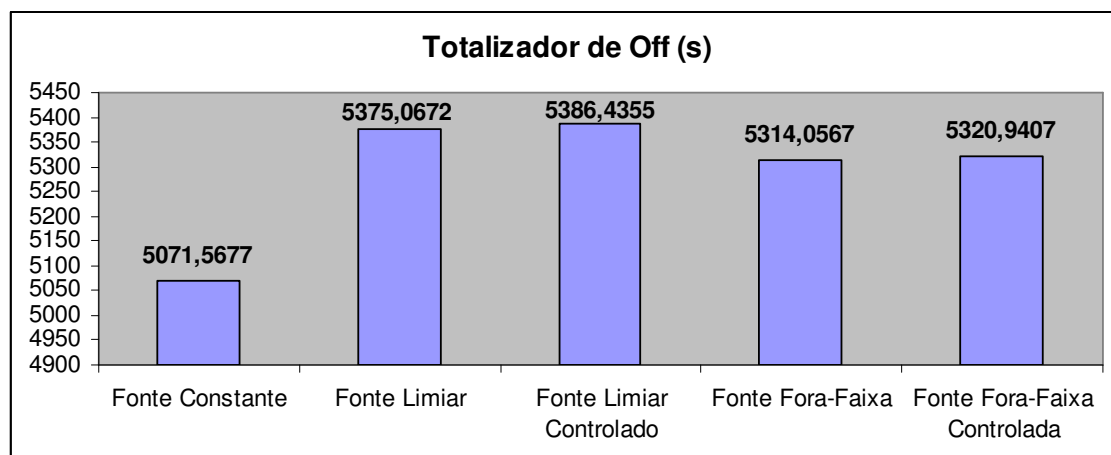


Figura 5.5: Comparação entre o totalizador de Off

Através da Figura 5.5 é possível perceber que as fontes orientadas a eventos possibilitam que os sensores permaneçam mais tempo em silêncio. Isso já era esperado, pois o totalizador de Off é inversamente proporcional ao Totalizador de On (Figura 5.4).

Também se pode notar que o parâmetro controle não altera significativamente o resultado das fontes que não apresentam este parâmetro. Afinal se comparados os tempos de Off das Fontes Fora-Faixa e Limiar com as respectivas fontes com o parâmetro controle, a alteração não equivale a um por cento (1%). Vale salientar que a mesma porcentagem foi obtida ao analisar a quantidade de intervalos On, apresentada na Figura 5.3.

Outro aspecto que assegura a confiança nos resultados é que o aumento nos totalizadores de Off, o que ocorre nas fontes que não tem o parâmetro controle, gera uma redução proporcional nos totalizadores de On.

5.4 ESCALONADORES PARA REDE CORPORAL

Em redes de sensores sem fio o escalonamento deve ser estudado em dois níveis: nível de sistema e a nível de nó. Em nível de sistema, o escalonamento é essencialmente um problema de Controle de Acesso ao Meio (MAC) e em nível de nó, o escalonamento determina qual pacote do buffer o nodo vai transmitir quando obtiver permissão [58] (p. 17).

Neste trabalho, será tratado somente o escalonamento em nível de sistema com a definição do MAC a ser utilizado e a sua análise.

Para o projeto de MAC para RSC devem ser levados em conta os principais fatores como evitar colisão de pacotes e retransmissão para economizar energia, simplicidade e o provimento de QoS [55], [107].

Os protocolos de controle de acesso se dividem em dois grupos: os baseados em contenção e os ordenados, sem contenção [117]. Nos protocolos baseados em contenção não há uma ordem de acesso ou priorização de algum nó. Assim, os nós podem iniciar a transmissão a qualquer momento, o que pode gerar colisão. Para evitar colisões os nós devem escutar o canal antes de iniciar a transmissão para verificar se o meio está livre [107].

Vale salientar que os protocolos baseados em contenção também são denominados protocolos de acesso aleatório [4]. Nesses protocolos um nó da rede pode enviar dados para o centralizador utilizando o CSMA/CA ou mecanismos de acesso Aloha. Para receber dados do centralizador da rede, o nó deve estar ativo e consultar o centralizador. A vantagem destes protocolos é que não há necessidade de efetuar consultas frequentes para identificar se há pacotes para serem transmitidos. A desvantagem é que o centralizador da rede não pode se comunicar com o nó, a menos que tenha sido convidado pelo nó para a troca de informações.

Por sua vez, nos métodos de acesso ordenados sem contenção, há uma sequência de atendimento aos nós para a transmissão, sendo que o nó deve aguardar sua vez para transmitir o pacote. Portanto, cabe ao nó centralizador organizar a disputa pelo canal. Assim, nos mecanismos de controle de acesso ordenados, o centralizador enviará permissões para os sensores transmitirem, bem como para sincronizar os dispositivos [122]. Além disso, compete ao centralizador a definição do ciclo de atendimento.

Ambos os mecanismos tem suas vantagens e desvantagens. Por exemplo, nos mecanismos de acesso aleatório, onde os nós disputam diretamente a vez para transmitir, pode haver colisões. A vantagem deste método é que não há necessidade de ficar ligando o rádio com frequência para verificar a chegada do pacote de permissão para a transmissão [122]. Já nos mecanismos de acesso ordenado, as colisões não ocorrem, pois o nó centralizador permite uma única transmissão a cada ciclo.

Para RSC, onde os sensores estão localizados numa área restrita (corpo humano), o uso do nó centralizado (ou nó sorvedouro) é mais conveniente. Isso se justifica pela simplicidade na comunicação, sendo capaz de evitar completamente as colisões, poupando energia. Deste modo, usando a configuração centralizada ou em estrela, o mecanismo de Controle de Acesso ao Meio (MAC) pode ser ordenado, impedindo que mais de um nó comece a transmitir pacotes ao mesmo tempo, evitando colisões de pacotes.

Um dos esquemas de acesso ordenado mais utilizado é o *Time Division Multiple Access* (TDMA) que permite, além das vantagens supracitadas, um tratamento quase em tempo real dos dados manipulados. Entretanto, o TDMA necessita uma sincronização de quadros, além do sincronismo de bits, o que traz ao esquema uma relativa complexidade.

Uma análise no padrão IEEE 802.15.4 para redes de sensores corporais foi realizada em [122]. Este padrão usa como mecanismo de controle de acesso o protocolo CSMA/CA, sendo concluído que com dez sensores a probabilidade do canal estar livre é quase 100%, bem como a de ocorrer colisões.

O algoritmo polling do tipo limitado foi usado por [2] para analisar o provimento de QoS em uma rede de sensores sem fio, sendo apresentado um modelo analítico e realizado testes através de simulação com o uso deste modelo.

Para [117] (p. 181), o polling é o método mais usado.

Um novo esquema de escalonamento baseado no polling exaustivo e limitado foi proposto por [76]. Com o esquema é possível melhorar a eficiência na transmissão, reduzir o consumo de energia, podendo ser utilizado para diferenciar serviço com prioridade, garantindo melhor QoS em RSSF.

O protocolo polling foi usado por [27], sendo demonstrado através de simulações que o novo modelo de cluster pode reduzir eficazmente o atraso e prolongar o tempo de vida da rede.

Um novo protocolo de acesso aleatório para aplicações médicas, com suporte a Qualidade de Serviço (QoS), e que se baseia no padrão IEEE 802.15.4a é proposto em [4]. Este padrão adota os mecanismos de acesso CSMA/CA e Aloha, sendo que a proposta possibilita um aumento drástico na taxa de transferência do tráfego prioritário.

A aplicabilidade do IEEE 802.15.4 em uma RSC foi analisada em [70], sendo relatado que melhorias significativas precisam ser realizadas neste MAC, tendo em vista que aplicações médicas ficam constantemente com os sensores no modo ativo.

Em [67] é relatado que o principal esquema MAC proposto na literatura para RSC é o padrão 802.15.4 com configuração estrela e autorização habilitada, que oferece consumo de energia muito baixo. No entanto, como o esquema não é projetado para aplicações de RSC, alguns inconvenientes têm sido apontados em [96] e, recentemente, muitos esquemas de protocolos MAC, específicos para RSC têm sido propostos [96], [4], [5], [17], [40], [66], [68], [72], [73], [81], [94], [123], [132], [133], [134]. Algumas propostas são variação do padrão 802.15.4 [5], [17] e [132], e outras são baseadas na técnica de acesso TDMA [40], [72], [73], [81], [94], [123], e [133]. Cada uma das propostas explora algumas características especiais com base em necessidades médicas. Por exemplo, em [133] e [73] para lidar com cargas leves e pesadas, em situações normais e de urgência,

um MAC é proposto baseado na estrutura do TDMA. Para garantir a QoS de uma RSC, em [4] é proposto um protocolo MAC baseado na técnica de acesso aleatório. Na proposta apresentada em [72], os batimentos do coração são utilizados para fins de sincronização do relógio.

Em todas as propostas supracitadas os nós devem ser acordados periodicamente para sincronizar o relógio do nó com o relógio do nó centralizador usando o sinal piloto (beacon).

Neste trabalho, o mecanismo de controle de acesso ao meio ordenado, denominado polling, que evita a necessidade de sincronização periódica é examinado. A ideia é usar o sistema de compartilhamento de tempo de conexão entre os terminais, semelhante ao computador mainframe, no início da computação. Neste mecanismo, o nó centralizador define um ciclo de atendimento dos nós. Baseado neste ciclo, o centralizador interroga individualmente cada nó para verificar se há pacotes para transmitir naquele instante. Se houver, o nó recebe permissão para iniciar a transmissão enquanto os outros aguardam sua vez. Assim, enquanto um nó transmite um pacote, os outros estão realizando suas atividades de monitoramento, e aguardando a sua vez para transmitir, podendo armazenar os pacotes gerados em buffer, se houver espaço disponível. Se não houver espaço, estes pacotes serão descartados [98]. Após analisar o último sensor, o ciclo de atendimento reinicia. Além disso, quando um nó não é inspecionado pode ser que esteja no estado desligado, economizando assim energia. Outra vantagem da técnica de acesso polling é que ela permite mudar o modo de investigação, dando diferente ordem de acesso aos sensores, possibilitando o emprego de QoS em RSC.

5.4.1 Polling

As disciplinas de escalonamento mais básicas, utilizadas em protocolos de polling são: exaustiva, gated e limitada [97]. A disciplina exaustiva consiste em transmitir até que o buffer da estação considerada esteja completamente vazio, passando em seguida para a próxima estação. Essa disciplina pode ter o problema de estabilidade, pois, se os pacotes chegam muito intensamente, pode ser que não se consiga completar o ciclo, examinando todos os buffers dos sensores, ou a

velocidade de inspeção tem que ser muito rápida, mas é mais eficiente. A disciplina gated consiste em transmitir todos os pacotes que estavam no buffer da estação no momento em que ela recebeu a permissão para acessar o canal. Porém, os pacotes que chegarem ao buffer após o início da transmissão, serão transmitidos apenas no próximo ciclo. Já a disciplina limitada consiste em respeitar um limite de transmissão, seja em tempo ou em número de pacotes, onde no momento em que o limite de transmissão é atingido, a estação corrente deixa de transmitir e a permissão de transmissão passa para a próxima estação, independente do estado do buffer da estação corrente.

Neste trabalho, a disciplina de polling utilizada é do tipo limitado, pois ao se examinar as filas, somente é transmitido o pacote da cabeça da fila, sendo, portanto, limitado a um pacote.

Além disso, de acordo com o funcionamento dos mecanismos de acesso ordenado, compete ao nó centralizador definir o ciclo de atendimento dos nós, sendo esta política adotada pelo mecanismo *polling*. Desta forma, os nós sensores são atendidos conforme este ciclo, verificando se os nós têm algum pacote para ser transmitido. Caso haja pacotes, o nó centralizador encaminha permissão para que o nó sensor inicie a transmissão. Caso outros sensores tenham pacotes para enviar neste mesmo tempo, se existir um buffer, estes pacotes podem ser armazenados, aguardando sua vez para transmitir. Caso não exista buffer, ou este já esteja cheio, o pacote é descartado. Ao final da transmissão do pacote, o próximo sensor do ciclo é interrogado. Se houver pacotes, a transmissão será realizada. Caso contrário, o sensor seguinte será questionado. Este ciclo se repete até que todos os nós sejam atendidos, reiniciando o ciclo [117].

O *polling* foi escolhido porque não necessita sincronização de quadros, como o TDMA requer e ainda pode realizar um tratamento de mensagens em tempo real para uma quantidade razoável de nós.

Além disso, ele pode ser simplificado usando o fato de que os sensores estão localizados muito perto do nó centralizador. Assim, em operação normal, o nó sorvedouro transmite um pacote enviando o número do nó a ser investigado, ou seja, ele envia uma autorização para um nó sensor transmitir os pacotes. Este pacote de autorização tem um cabeçalho suficiente para a sincronização bit a bit e de quadros de um nó. Se um nó tem pacotes para transmitir, reconhece o número do nó e

começa a transmissão. Após a transmissão o nó sensor espera o reconhecimento para identificar a necessidade de retransmissão. Se um nó não tem pacote para transmitir, o transceptor pode ficar no estado desligado e só mudar para o estado ativo se houver pacotes para transmitir. O nó centralizador reconhece que um nó está no estado desligado após a transmissão do pacote de autorização e depois de ter esperado por um tempo. Se o pacote de dados não veio do nó investigado, o nó sorvedouro entende que o nó não tem pacote para transmitir e vai para o próximo nó do ciclo de atendimento. Assim, neste protocolo o nó sorvedouro faz quase todas as funções de comunicação, deixando para o nó sensor apenas a função de transmissão de pacotes.

Com o acesso ao canal de entrada organizado, os pacotes que chegam ao nó sorvedouro devem ser enfileirados no buffer de saída para serem retransmitidos para a Internet ou uma rede dedicada, onde os dados são processados e armazenados. Para realizar este processo de enfileiramento são utilizados algoritmos de escalonamento.

Na Figura 5.6, um exemplo do funcionamento do polling limitado adotado neste trabalho pode ser visualizado. O nó sorvedouro, em funcionamento normal, define um ciclo para atender os nós. Com base neste ciclo, o nó sorvedouro interroga cada sensor individualmente para verificar se existem pacotes para transmitir. Se houver, o nó sensor recebe uma permissão para iniciar a transmissão, enquanto os outros esperam sua vez. Assim, enquanto um nó sensor transmite pacotes, os outros estão realizando suas atividades de monitoramento, e esperando sua vez para transmitir, podendo armazenar os pacotes gerados no buffer. Após a transmissão de dados de um nó sensor, este pode alterar o status para o estado dormir, economizando assim energia.

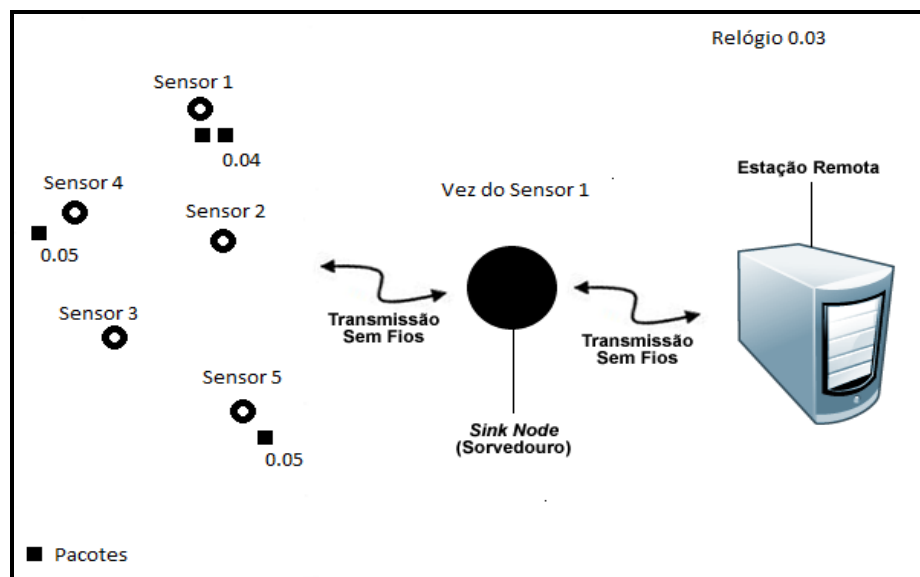


Figura 5.6: Funcionamento do mecanismo polling.

Através da Figura 5.6, pode ser visualizado que o sensor 1 será o primeiro a ser atendido, pois apenas este sensor possui pacote para transmitir neste instante. Após, o sensor 2 será verificado. Como o sensor 2 e o sensor 3 não têm pacotes para serem transmitidos, o próximo sensor é interrogado. Como o sensor 4 possui pacote para ser enviado, antes que isso seja realizado, é verificado se em algum outro sensor há pacotes com menor tempo de chegada. Como não há, o primeiro pacote do sensor 4 será enviado. Uma vez que o primeiro pacote do sensor 4 e do sensor 5 tem o mesmo tempo de chegada, e o servidor está ocupado processando o pacote do sensor 4, o pacote do sensor 5 terá que ser armazenado em buffer, se existir, ou será descartado. Caso haja buffer e tiver posição disponível, o tempo de fila do pacote terá que ser contabilizado. Para isso é subtraído o tempo de chegada do valor do relógio.

Este é o funcionamento do mecanismo implementado neste trabalho, sendo que os pacotes gerados pelas fontes propostas e descritas anteriormente, são armazenados em vetores. Vale ressaltar que cada fonte armazena os pacotes em um vetor distinto, independente do número de sensores de cada fonte.

Após a geração das fontes, os dados são concatenados em um único vetor, ordenados por tempo de chegada.

Há também um buffer para cada sensor, sendo que as informações são armazenadas em um vetor único. É importante salientar que o código desenvolvido é flexível, sendo possível que os sensores tenham tamanhos de buffer diferentes. Então, em ambas as situações há necessidade de identificar a posição do vetor onde se iniciam as informações de determinada fonte e sensor.

Além disso, para a geração das simulações é preciso informar o descarte estatístico, ou seja, após quantos pacotes gerados serão contabilizadas as informações para fins de relatório. Isso é realizado para que o sistema possa entrar em equilíbrio.

Após alimentar o programa com as informações, bem como inicializar todas as variáveis, inicia-se o processo de polling, conforme apresentado na Figura 5.6. Como para o funcionamento do polling é preciso definir e seguir uma ordem de atendimento, inicialmente é verificado se há pacotes na fila do sensor que tem permissão para transmitir. Se houver, este pacote será enviado. Deste modo, é verificado se o tempo de chegada do pacote que está na fila é menor que o relógio. Se for, o tempo de fila deste pacote é calculado, subtraindo o tempo de chegada do valor armazenado na variável relógio.

Como mencionado na revisão sobre fontes para RSC, o tráfego é influenciado pelo fenômeno monitorado, bem como pelo modo de operação dos sensores. Desta forma, o tempo de chegada dos pacotes deve ser recalculado ao chegar ao nó sorvedouro. Vale reforçar que para os pacotes que são descartados no nó sensor, este processo não é realizado. Isso ocorre, pois, ao contrário das redes cabeadas, os pacotes são gerados em um tempo, esperam para serem transmitidos, sendo enviados em outro tempo. Então no cálculo do tempo de chegada tudo isso é considerado e este equivale ao tempo que o pacote foi gerado adicionado ao tempo que esperou no polling e o tempo que levou do nó sensor até o sorvedouro. O tempo de envio do pacote do nó sensor até o sorvedouro é obtido somando o intervalo de On deste pacote ao relógio.

Após encaminhar o pacote que estava na fila, deve-se verificar se no intervalo transcorrido para o envio do pacote, outros sensores geraram tráfego. Em caso afirmativo, o mesmo estará no buffer, caso haja, ou foi descartado. Para este processo há uma estrutura de repetição que verifica sensor por sensor, analisando se o tempo de chegada dos pacotes é menor que o relógio.

Caso o sensor da vez tenha pacotes para transmitir, mas que não estão na fila, é preciso verificar se o tempo de chegada é maior do que o relógio. Caso seja, deve-se verificar se há pacotes de outros sensores com tempos de chegada inferior, sendo que estes devem ser tratados primeiro. Caso não haja nenhum pacote com tempo de chegada menor que o pacote atual, este será transmitido.

5.5 ANÁLISE DE UMA REDE CORPORAL COM ESCALONADOR POLLING

As fontes propostas, o escalonador polling e o FIFO foram implementados e integrados na plataforma de simulação desenvolvida usando a linguagem C++ Builder.

A Figura 5.2 mostra o ambiente que será analisado.

Pode ser visto na Figura 5.2 que o corpo humano foi dividido em sete regiões: a cabeça (região 1), tórax (região 2), dois membros superiores (regiões 3 e 4), abdômen (região 5) e dois membros inferiores (regiões 6 e 7), seguindo o esquema proposto em [110]. Assim, sete sensores foram inseridos no corpo humano, de acordo com o que foi proposto em [71].

Os nós sensores transmitem a sua informação para o sorvedouro que foi inserido no meio dos nós sensores. O mecanismo polling é colocado no nó sorvedouro.

A partir do sorvedouro, os dados são transmitidos para uma porta de entrada, que pode ser um dispositivo móvel, como mostrado na Figura 5.2. O nó sorvedouro também contém o escalonador FIFO. Do telefone, a informação é enviada para um servidor.

Diferentes tipos de fontes, usando os parâmetros listados na Tabela 5.5, são colocados em partes distintas das regiões do corpo humano, de acordo com a Tabela 5.11.

Tabela 5.11: Cenários Simulados

	Cenário 1	Cenário 2	Cenário 3
Região 1	Constante	Constante	Constante
Região 2	Constante	Constante	Limiar
Região 3	Constante	Constante	Limiar Controlado
Região 4	Constante	Limiar	Limiar Controlado
Região 5	Constante	Limiar Controlado	Fora-Faixa
Região 6	Constante	Fora-Faixa	Fora-Faixa Controlada
Região 7	Constante	Fora-Faixa Controlada	Fora-Faixa Controlada

Na Tabela 5.11, pode-se visualizar os três cenários considerados nas simulações, os quais visam verificar o desempenho da RSC estudada. No Cenário 1, a fonte constante é usada em todas as partes do corpo para estudar o desempenho do sistema polling ao lidar com o tráfego de pacotes mais pesado, visto que a fonte gera constantemente os pacotes. No segundo cenário, fontes mistas foram configuradas, predominando a fonte constante. O terceiro cenário é misturado com os diferentes tipos de fontes propostas.

Cada fonte foi simulada com 10.000 pacotes, e os tamanhos de buffer utilizados foram um (1), três (3), cinco (5) ou mil (1000) posições. A escolha destes valores justifica-se pelo fato de que os nós possuem pouca memória, sendo que a última situação equivale a um buffer fictício, devido à energia limitada dos nós e pode ser considerado infinito, dependendo da taxa de geração dos pacotes. Uma vez que não existem variações frequentes nas medições de sinais fisiológicos, tais como temperatura e pressão, algumas perdas podem ser aceitas sem impacto para o sistema. Além disso, em [71] foi utilizado buffer de 512B, 1KB, 4KB ou 8KB, o que justifica os tamanhos de buffer adotados na análise.

Para as fontes que usam o parâmetro controle, um pacote de sinalização que o nó sensor está ativo é enviado a cada dez pacotes não transmitidos.

O mesmo enlace físico de saída de 250 Kbps usado no MicaZ [23] é adotado. Os dados para análise estatística são coletados depois de eliminados os primeiros 2000 pacotes, para garantir que o equilíbrio estatístico seja alcançado.

Para analisar o comportamento da RSC apresentada na Figura 5.2, os três cenários mostrados na Tabela 5.11 foram considerados. O objetivo é analisar a perda de pacotes e o tempo de fila nos nós sensores e no escalonador FIFO usando o mecanismo de controle de acesso ao meio polling. Além disso, o objetivo é verificar também a influência das diferentes configurações dos sensores no desempenho da RSC.

A Figura 5.7 mostra os resultados das simulações para o Cenário 1, consoante a perda de pacotes.

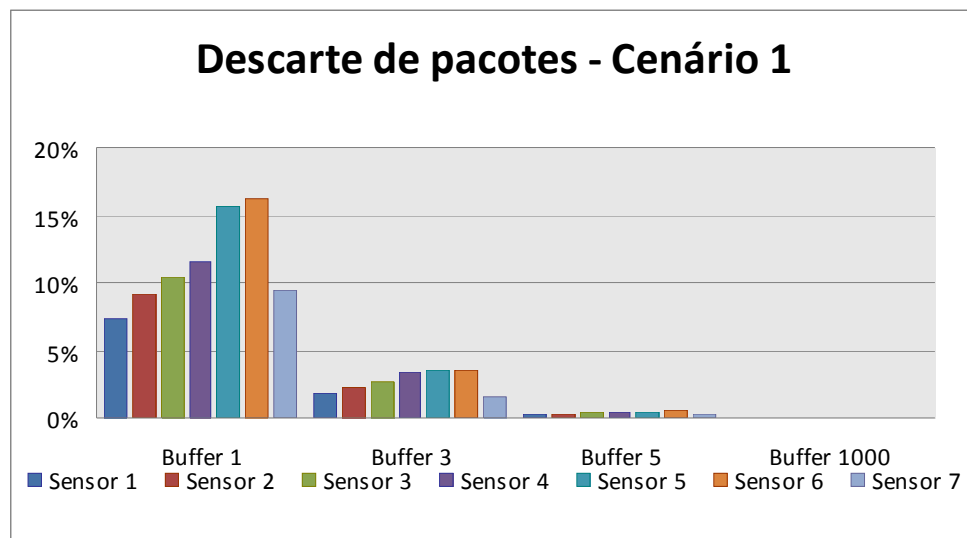


Figura 5.7: Perda de pacotes no primeiro cenário

Como pode ser visto na Figura 5.7 as perdas de pacotes em todos os nós são muito altas, variando de 7% no nó 1 para 16% no nó 6, para o buffer com uma posição, porque todos os pacotes gerados são transmitidos. No entanto, com o aumento de tamanho do buffer para três posições, as perdas de pacote tornam-se razoáveis atingindo o máximo de 4%. Para tamanho do buffer com cinco posições, as perdas são inferiores a 1%, e para mil posições não há perda de nenhum pacote.

A perda de pacotes usando Fonte Constante pode não ser crítica, porque as fontes estão constantemente enviando pacotes e alguns pacotes perdidos podem ser interpretados no servidor utilizando alguma técnica de interpolação.

Na Figura 5.8 é mostrado o tempo médio de fila dos pacotes.

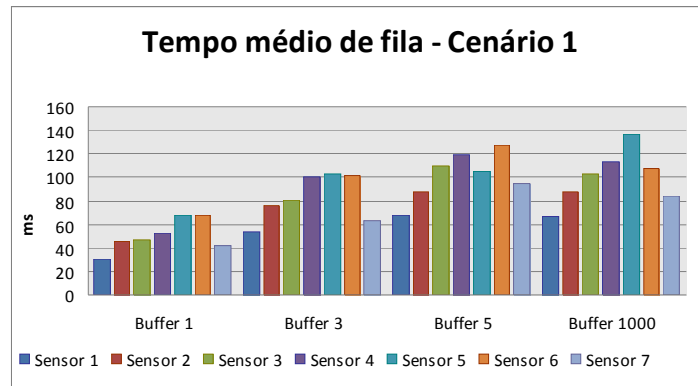


Figura 5.8: Tempo de fila dos pacotes no primeiro cenário

O tempo mínimo de espera é de cerca de 30 ms, no nó 1, para buffer de uma posição e o máximo é de aproximadamente 140 ms, no nó 5, para um buffer de mil posições, como pode ser visto na Figura 5.8. Os tempos de espera não são longos e são apropriados para processamento em tempo quase real.

Nas simulações de todos os cenários, não é considerado o tempo de percurso necessário, após a transmissão de um pacote, para mover o controle de um nó para outro nó. Este tempo, no caso de RSC é pequeno e pode ser considerado um valor constante. O tempo de propagação de um nó para o nó sorvedouro também não é considerado.

A Figura 5.9 mostra as estatísticas do nó sorvedouro quando o escalonamento FIFO é usado para transportar dados até a rede pública.

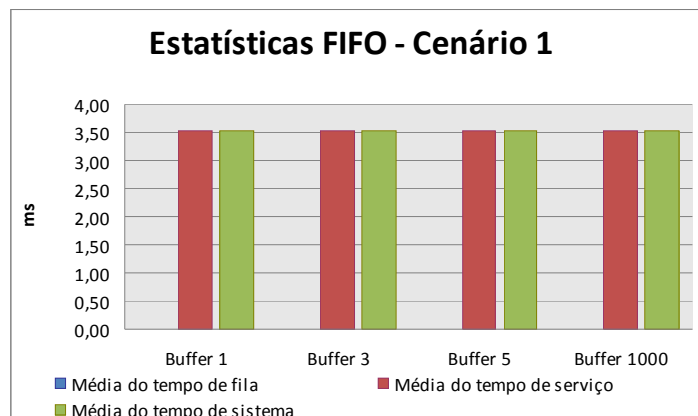


Figura 5.9: Tempo de fila, serviço e sistema no nó sorvedouro com o algoritmo FIFO e Cenário

Na Figura 5.9 o tempo de serviço, que corresponde ao tamanho do pacote dividido pela capacidade do link de saída de 250 kbps, o tempo de fila e o tempo de sistema, que é a soma dos tempos de fila e de serviço, são mostrados. Como o mecanismo *polling* organiza o acesso dos pacotes e a capacidade do canal é suficientemente rápida para transmitir um pacote, chega apenas um pacote por vez no FIFO. Assim, não ocorre perda de pacotes e nenhum pacote espera na fila FIFO.

A Figura 5.10 mostra os resultados das simulações para o Cenário 2, considerando a perda de pacotes.

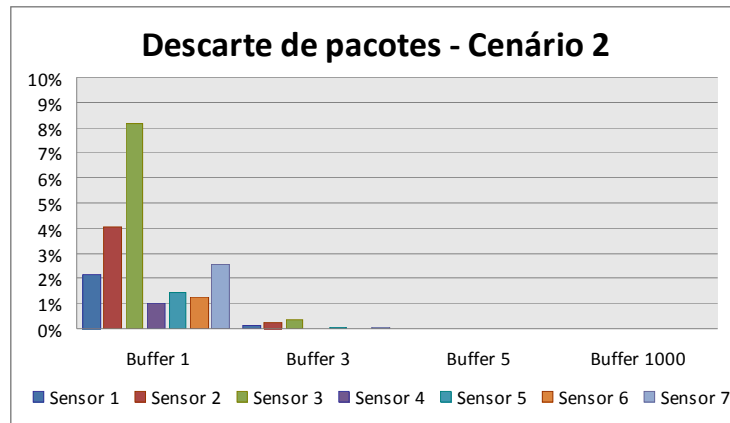


Figura 5.10: Descarte de pacotes do segundo cenário

No segundo cenário, os tráfegos são gerados em três sensores usando a fonte constante, na qual todos os pacotes gerados são transmitidos, o que pode ser visto nos nós 1, 2 e 3, onde a perda de pacote é mais significativa, atingindo 8,14% no nó 3. Este fato é justificado considerando-se que todos estes três nós foram configurados com os parâmetros mostrados na Tabela 5.5, e, assim, os tempos de chegada dos pacotes são similares. Como o serviço é cíclico, começando com o primeiro nó, seguido pelos nodos 2, 3 e assim por diante, o nó 3 tem de esperar os dois nós serem atendidos primeiro para iniciar sua vez, tendo assim maior perda, o que é confirmado na Figura 5.10, com uma posição de buffer. Nos outros nós, com uma posição de buffer, a perda alcançou, no máximo, 2,50% no nó 7, que utiliza a fonte Fora-Faixa Controlada para a geração dos pacotes.

Comparando estes valores com as simulações que usam tamanhos maiores de buffer, o descarte diminuiu consideravelmente, e quando o tamanho do buffer é fixado em cinco ou mil posições, a perda de pacotes não aconteceu em nenhum dos nós.

A Figura 5.11 mostra a porcentagem de pacotes não transmitidos em função da restrição imposta para as fontes.

Devido ao uso da Fonte Constante nos nós 1, 2 e 3, não existem pacotes rejeitados nesses nós como pode ser visto na Figura 5.11. Em outros nós, os quais usam fontes com critérios restritivos, o percentual de rejeição é superior a 50% em todos os nós, sendo a maior no nó 4, que usa a Fonte Limiar.

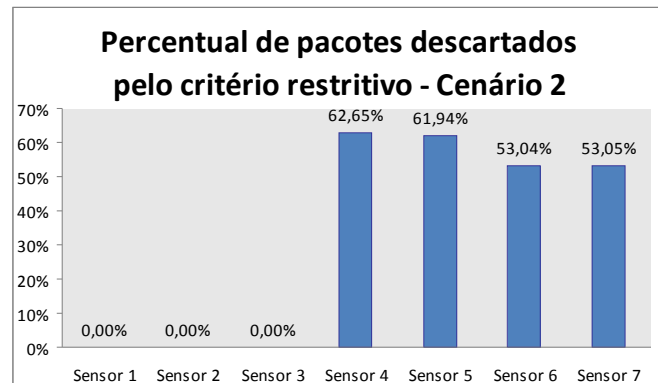


Figura 5.11: Percentual de pacotes descartados, no Cenário 2, pelos critérios restritivos

Pode ser observado na Figura 5.11 que os valores usados não são afetados pelo tamanho do buffer porque as rejeições de pacotes são feitas usando o algoritmo restritivo antes da fila no buffer do nó.

Também pode ser visualizada a importância da aplicação do algoritmo restritivo para poupar energia, considerando o elevado consumo de energia em uma transmissão de pacotes.

A Figura 5.12 apresenta os resultados das simulações do tempo de fila no buffer para o Cenário 2. A mesma influência do atendimento *polling* da Figura 5.10 é também observada neste caso, com o nó 3 tendo mais tempo de espera para transmitir os seus pacotes. Em outras fontes, que

não usam a fonte constante, os tempos de espera são menores devido à geração de pacotes ser controlada.

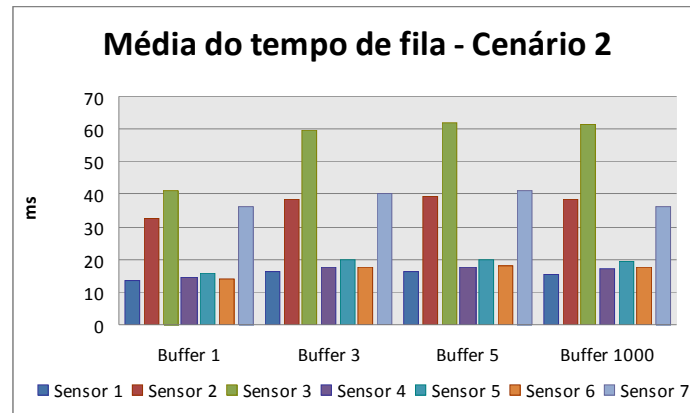


Figura 5.12: Tempo de fila dos pacotes no segundo cenário

Quando o tamanho do buffer é aumentado, os tempos de espera são maiores e têm um comportamento semelhante, demonstrando que não é necessário grandes tamanhos de buffer, mas um buffer com 3 posições é suficiente para que a perda de pacote seja baixa, possibilitando a transmissão de quase todos os pacotes gerados.

Na Figura 5.13 os tempos de serviço, de fila e, de sistema no sorvedouro usando o escalonador FIFO são mostrados.

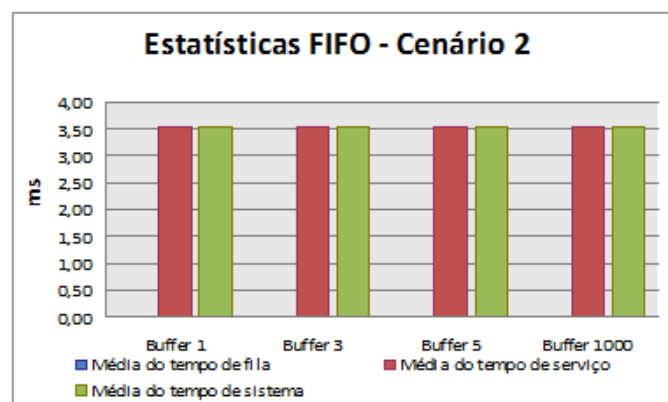


Figura 5.13: Tempo de fila, serviço e sistema no sorvedouro com o algoritmo FIFO no Cenário 2.

Os resultados são os mesmos obtidos na Figura 5.9 e a conclusão é também a mesma. É claro para todos os cenários, que o tamanho do buffer de uma posição para o nó sorvedouro é suficiente para atender toda a demanda.

A Figura 5.14 mostra a perda de pacotes para o Cenário 3 em função dos tamanhos de buffer.

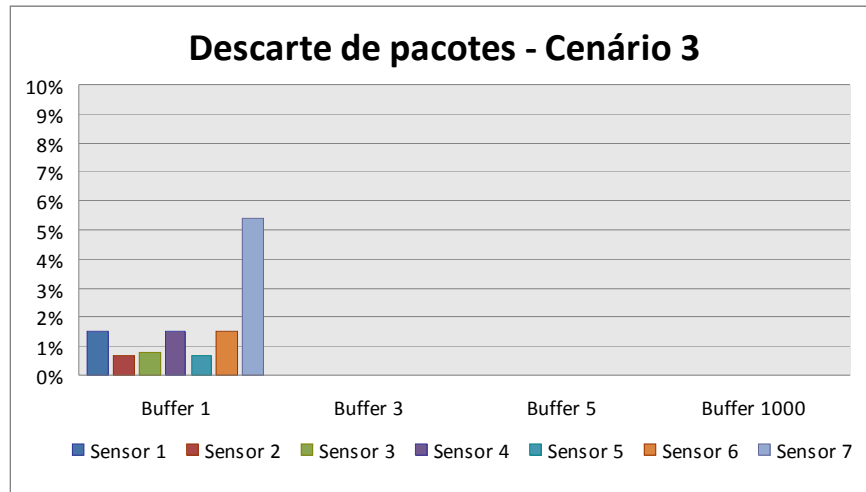


Figura 5.14: Descarte de pacotes no terceiro cenário

Neste Cenário 3, como o número de sensores com a Fonte Constante é restrita ao sensor 1, o número de pacotes transmitidos diminui. Por conseguinte, a perda de pacote é reduzida. Apenas para efeito de comparação, em média, 46.435 pacotes são transmitidos no Cenário 2, enquanto no Cenário 3 este valor é 35.747 pacotes, com a redução de 10.688 pacotes. No Cenário 2, a maior perda foi de 814 pacotes no Sensor 3, enquanto no Cenário 3 este fato ocorreu no sensor 7, chegando a 255 pacotes perdidos, considerando-se uma posição de buffer. Entretanto, ao usar cinco e mil posições de buffer, nos Cenários 2 e 3 não há perdas de pacotes.

Comparando os Cenários 2 e 3 em relação ao descarte de pacotes, os nós 1, 2 e 3, que utilizam a fonte constante, têm maiores perdas no Cenário 2, como era esperado. Além disso, como o serviço é cíclico e começa no nó 1, este nó torna-se o mais favorecido dos três nós, com descarte mais baixo. No terceiro cenário, uma vez que uma diversidade de fontes é utilizada, principalmente aquelas que usam o parâmetro de controle, o nó que produz o maior descarte é o sensor 7. Este nó é

o último a ser atendido no ciclo do *polling*. Assim, ele espera mais tempo para transmitir e pode não transmitir nenhum pacote em um dado ciclo, descartando todos os pacotes porque eles estão dentro do intervalo pré-definido ou abaixo de um limiar.

A Figura 5.15 mostra o percentual médio de pacotes não enviados devido ao critério restritivo aplicado nas fontes para o Cenário 3.

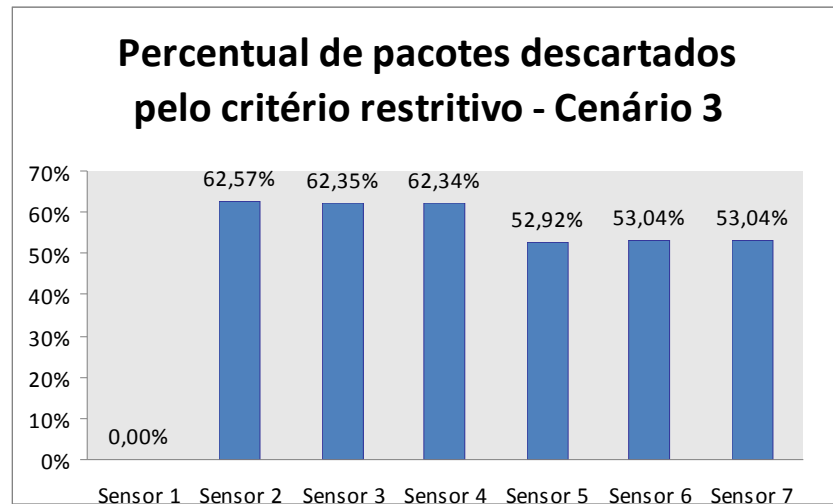


Figura 5.15: Percentual de pacotes descartados, no Cenário 3, pelo critério restritivo

Pode ser visto na Figura 5.15 que não há pacote rejeitado no nó 1, devido à utilização da fonte constante como também é observado na Figura 5.11. No entanto, os percentuais de rejeição são altos em outros nós, os quais utilizam fontes com restrições para enviar pacotes, como ocorreu no Cenário 2. As Fontes Limiar e Limiar Controlado têm percentuais mais elevados de rejeição que variam de 62,57% para 62,34%, como pode ser visto nos nós 2, 3 e 4. A Figura 5.15 mostra que nas Fontes Fora-faixa e Fora-faixa Controlada as perdas também são significativas atingindo cerca de 53% para todas as três fontes, como pode ser observado nos nós 5, 6 e 7.

O consumo de energia também pode ser observado neste Cenário 3, evitando a transmissão de pacotes, embora alguma energia seja consumida para o processamento do algoritmo restritivo.

Na Figura 5.16 os tempos de fila para o terceiro cenário são ilustrados.

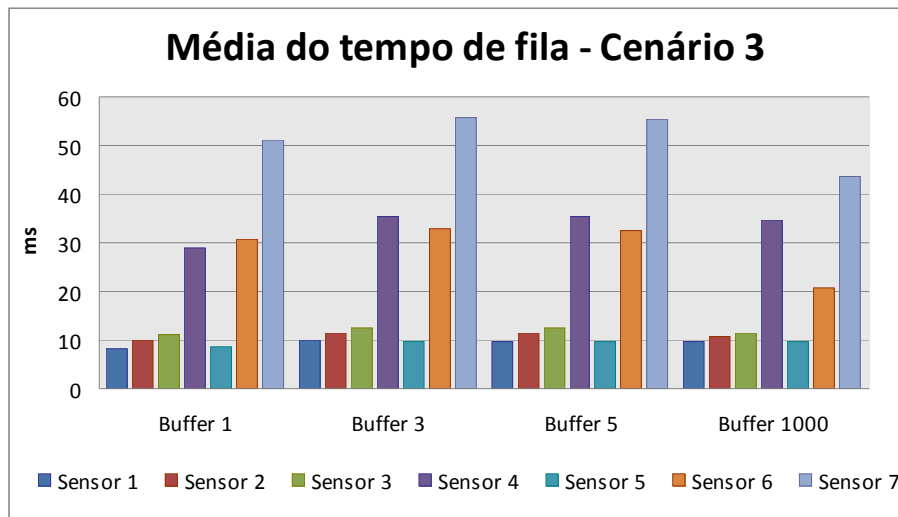


Figura 5.16: Tempo de fila dos pacotes no terceiro cenário

Ao reduzir o número de pacotes transmitidos, os tempos de fila são reduzidos no Cenário 3 em comparação com os Cenários 1 e 2, conforme esperado. A redução é significativa na maior parte dos nós, como pode ser verificado na Figura 5.16. Nos Cenários 1 e 2 os tempos médios de fila são 50,43 ms e 23,87 ms, respectivamente, enquanto no Cenário 3 é 21,21 ms considerando buffer com uma posição. Além disso, os casos mais graves para os Cenários 1 e 2 ocorreu no nó 5 (ou 6) e no nó 3, com tempos de espera de 68 e 40,89 ms, respectivamente, enquanto no Cenário 3, o maior tempo de fila é 50,93 ms no nó 7 usando a fonte Fora-faixa Controlada.

A utilização de diversificadas fontes de tráfego é benéfica para o tempo de fila e perda de pacotes na maioria dos nós. Além disso, a predominância do tipo de fontes orientada a eventos também é importante. Para aplicações de RSC um máximo de três posições de buffer pode atender as expectativas de QoS para o descarte de pacotes e tempo de fila.

Por outro lado, observa-se que, com buffers de cinco e mil posições, existe semelhança entre os Cenários 2 e 3 nos tempos de fila, porque não existem variações significativas nos valores apresentados.

A Figura 5.17 inclui os resultados das simulações para os tempos de serviço, de fila, e do sistema no nó sorvedouro, utilizando o escalonador FIFO para enviar dados para a rede pública.

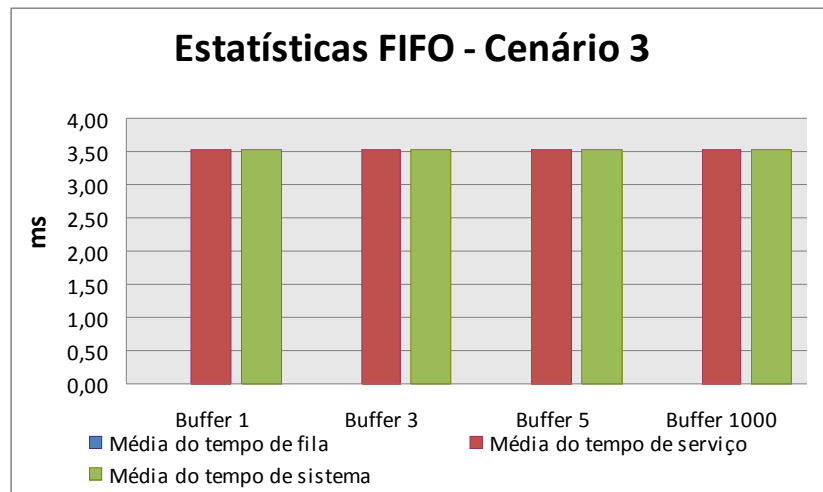


Figura 5.17: Tempo de fila, serviço e sistema no sorvedouro com o algoritmo FIFO para o Cenário 3

Na Figura 5.17 pode ser confirmado que o mecanismo *polling* tem a função de controlar a admissão de tráfego para que os tempos de serviço e de fila não sofram variações por parte dos tipos de fontes utilizados nos nós sensores. Assim, os resultados dos Cenários 1, 2 e 3 são iguais em relação aos tempos de espera, de serviço, e do sistema. Além disso, o tamanho do buffer utilizado nos nós não altera as estatísticas no nó sorvedouro nos três cenários, como pode ser visto nas Figuras 5.9, 5.13 e 5.17. O tempo de serviço ou o tempo de sistema é de 3,5 ms em todos os cenários analisados.

5.6 CONCLUSÕES DO CAPÍTULO

Neste capítulo abordaram-se conceitos e o funcionamento das redes de sensores corporais, apresentando os benefícios do monitoramento remoto, bem como os problemas que estes sistemas apresentam. Estes estão diretamente relacionados ao consumo de energia em função das restrições dos nós sensores, sendo que, conforme relatado, o maior consumo refere-se à comunicação dos nós.

Para auxiliar no processo de comunicação e conseqüentemente reduzir o consumo de energia, mecanismos de controle de acesso ao meio, bem como modelos de fontes de tráfego para

redes corporais foram analisados. No que tange aos escalonadores para redes corporais, conforme apresentado na literatura, o mecanismo de controle de acesso *polling* é uma boa alternativa. Consoante as fontes de tráfego, o modelo On/Off pode ser considerado adequado para simular o tráfego de redes de sensores corporais.

Através do programa Fonte On/Off Constante, desenvolvido em [115], o qual gera um pacote a cada intervalo On, foram feitas adaptações no código dessa Fonte para atender ao novo propósito. Além disso, uma pesquisa exaustiva na literatura foi realizada com o intuito de obter os parâmetros necessários para a geração do tráfego que emule o comportamento de uma RSC. Com isso os dados apresentados na Tabela 5.5 foram utilizados.

Visando testar o programa, simulações foram feitas com 10000, 20000 e 30000 pacotes. Os resultados obtidos comprovaram a convergência para os parâmetros estabelecidos, demonstrando o correto funcionamento da aplicação.

Devido a quantidade de intervalos On gerada pela Fonte On/Off Constante percebeu-se que ela atende ao modo de monitoramento contínuo, ocasionando muito tráfego na rede. Desta forma, alterações foram realizadas no programa Fonte On/Off Constante, para emular o comportamento dos sensores trabalhando em modo orientado a evento, gerando assim quatro novas fontes.

Desta forma, cinco tipos de fontes foram propostas, todas baseadas no modelo On/Off. A Fonte On/Off Constante é a única que encaminha as informações continuamente. As demais são fontes orientadas a evento, no qual apenas são transmitidas as informações que satisfizem a condição estabelecida. Além disso, para facilitar o gerenciamento do status dos sensores, em duas fontes orientadas a evento, uma mensagem é enviada após um determinado número de pacotes não transmitidos, independente se o requisito para o envio dos dados seja atendido ou não. Este parâmetro é chamado controle.

Um resumo da característica das fontes propostas é apresentado na Tabela 5.12.

Tabela 5.12: Característica das fontes propostas

Fonte On/Off Constante	Envia todos os pacotes que foram gerados.
Fonte On/Off Limiar	Envia somente pacotes que carregarem uma informação acima de determinado valor.
Fonte On/Off Limiar Controlado	Envia somente pacotes que contenham informação acima de um determinado limiar ou que tenha sido descartado um número pré-definido de pacotes.
Fonte On/Off Fora-Faixa	Envia somente pacotes que carregarem uma informação que esteja fora de determinado intervalo.
Fonte On/Off Fora-Faixa Controlada	Envia pacotes que atendam ao critério da Fonte On/Off Fora-Faixa . Fora isso, apenas se um número superior de pacotes não tenha sido transmitido, o pacote será enviado independente da informação.

Mediante a comparação entre as fontes, constatou-se que a Fonte On/Off Limiar transmite aproximadamente 37% dos pacotes sensorizados. Já com a introdução do parâmetro controle o aumento gerado não chega a um por cento (1%).

A Fonte On/Off Fora-Faixa transmite aproximadamente 47% dos pacotes e, se inserido o parâmetro controle nesta fonte, a variação é ínfima, não chegando a um por cento (1%).

Com isso percebe-se que o parâmetro controle deve ser implementado nas fontes, tendo em vista que o mesmo permite simplificar a implementação do mecanismo de Controle de Acesso ao Meio (MAC), pois o sorvedouro será constantemente informado sobre o status do sensor.

Além disso, é possível concluir que usar fontes orientadas a evento é o ideal para aplicações em RSC. Afinal os pacientes estarão sendo monitorados constantemente, porém a energia será utilizada somente com transmissões necessárias.

Após a análise e implementação das fontes propostas, o simulador para escalonamento FIFO foi modificado, alterando as fontes de tráfego e incorporando o mecanismo de controle de acesso *polling*. Através destas adaptações, o simulador foi utilizado para analisar o tempo de fila e descarte de pacotes nos sensores, com diferentes tamanhos de buffer, bem como os tempos de fila, serviço e

sistema no sorvedouro. Uma vez que o nó sensor para RSC necessita poupar energia, também foi estudado o tamanho do buffer mínimo para manter os pacotes antes das transmissões.

O ambiente do corpo humano proposto para o estudo consistiu em sete sensores colocados em diferentes partes do corpo, formando uma topologia em estrela, tendo o sorvedouro no centro da rede. Neste ambiente, três cenários foram propostos. O intuito foi verificar o impacto que os diferentes modelos de fontes geram no que se refere ao descarte de pacotes e tempo de fila, além de analisar o comportamento do mecanismo *polling* no que se refere ao controle de acesso e transmissão das informações. O primeiro cenário utiliza uma configuração com fonte constante em todos os nós, enquanto que no segundo cenário três fontes constantes são misturadas com outros tipos de fontes. No último cenário cinco fontes são misturadas em diferentes partes do corpo.

Os resultados das simulações para o Cenário 1 em relação à perda de pacote, considerando apenas uma posição de buffer no nó sensor mostraram perdas elevadas que variam de 7% a 16%. As perdas tornaram-se razoáveis para três ou maiores tamanhos de buffer.

No Cenário 2, as perdas de pacotes nos três sensores que usam a fonte constante também são elevadas, atingindo mais de 8% em um dos nós e cerca de 2% no nó mais favorecido pelo sistema *polling*. No entanto, essas perdas não podem ser críticas para fontes constantes, porque elas estão enviando frequentemente os pacotes para que alguma informação perdida possa ser interpretada no servidor final utilizando alguma técnica de interpolação. Mas estas fontes devem ser usadas com cuidado devido à elevada intensidade de geração de pacotes. Para os outros quatro nós sensores, do Cenário 2, as perdas são menores variando de 1% a cerca de 2%. Mas, nestes casos, as perdas podem ser críticas porque os sensores já estão fazendo algum tipo de seleção de dados. Para superar esta situação, os resultados mostraram que, para um buffer de três posições, as perdas são quase insignificantes.

Para o Cenário 3, numa situação mista de fontes, as perdas são mais controladas, atingindo no pior caso cerca de 5% com buffer de uma posição e nenhuma perda ocorre no caso de buffers com três ou mais posições.

Os resultados da simulação referentes aos tempos médio de fila nos sensores apresentaram baixos tempos de espera para os Cenários 1, 2 e 3, variando de 8 a 140 ms, considerando todos os tamanhos de buffer. Não se consideraram os tempos de propagação e percurso nas simulações, porque são valores pequenos e constantes. Assim pode-se concluir que o esquema de acesso *polling* é adequado para aplicações em tempo quase real, quando cenários similares são considerados.

A partir dos resultados obtidos percebe-se que o mecanismo de controle de acesso *polling* atende aos requisitos propostos, organizando o acesso ao canal de saída adequadamente, de tal forma que não há descarte no sorvedouro que usa o escalonador FIFO. Isso ocorre mesmo tendo o sorvedouro uma única posição de buffer. Além disso, não há fila de espera se a taxa de geração for relativamente baixa, sendo esta similar ao do sensor Mica2Dot. Contudo, se esta taxa for aumentada, a fila de espera passa a ser aproximadamente de 0,0004 s. Outrossim, independente das fontes usadas para a geração de tráfego, não há variações nos tempos de fila, serviço e sistema, se forem usados os parâmetros apresentados na Tabela 5.5 para a geração das fontes.

Simulações foram realizadas com tamanho de pacote de 32 bits, tempo de Off de 0,00799 s, tempo de On de 0,00088 s e taxa de pico de 36045 bps. Com estas configurações para a geração das fontes, aumentando o tamanho do buffer também aumenta o tempo médio de fila, sendo este aproximadamente a metade do tempo médio de fila com as configurações de pacotes apresentadas na Tabela 5.5. Quanto ao descarte de pacotes, este também reduz drasticamente, assim como ocorre nos cenários analisados.

Mediante os resultados apresentados, percebe-se que a Fonte On/Off Constante deve ser evitada, pelo fato de gerar grande tráfego na rede. Além disso, diversificar as fontes usadas é uma estratégia ótima para reduzir o descarte e o tempo de fila. Também se pode perceber que o mecanismo *polling* é uma boa alternativa para o controle de admissão. No entanto, comparações com outros mecanismos de acesso baseados em 802.15.4 ou em padrão mais recente 802.15.6 devem ser realizadas para melhor avaliação do esquema *polling* proposto.

6 CONCLUSÕES

Os dois objetivos principais desta tese foram: em primeiro, analisar e propor escalonadores que satisfaçam a QoS das redes convergentes e em segundo, propor escalonador conveniente para redes de sensores corporais.

Para a análise de escalonadores para redes convergentes foi desenvolvida uma plataforma de simulação em C++ Builder em que várias partes da rede foram implementadas. Uma das partes foi o desenvolvimento de diversas fontes de tráfego para redes convergentes, todas baseadas em modelo On/Off. Foram, também, desenvolvidos os escalonadores FIFO, DRR e DRR de banda efetiva, sendo este último uma proposta inédita na literatura. Além dessas partes, a plataforma desenvolvida constou de buffers e um enlace de saída. Com esta plataforma foi possível verificar o impacto dos tráfegos multimídia em escalonadores como FIFO e DRR.

Em relação as fontes On/Off exponenciais para redes convergentes, analisadas no capítulo 3, constatou-se que a Fonte 2.0 gera maior ociosidade no servidor, devido ao tempo de chegada entre um pacote e outro ser maior. No entanto, se as chegadas forem mais próximas, o que ocorre na Fonte Fixa e Fonte 1.0, pode ocorrer maior perda de pacotes.

Consoante aos escalonadores para redes convergentes foi possível comprovar que o escalonador FIFO, embora não efetue nenhum tipo de priorização, é muito adequado quando o tráfego é restrito a dados. Contudo, para aplicações em tempo real, como o caso de tráfego de voz e vídeo, este escalonador já não se comporta de maneira adequada. Quanto ao tráfego de voz há um crescimento considerável no tempo de sistema. Porém isso é ainda maior no que se refere ao tráfego de vídeo, o qual não admite um grande número de usuários no sistema.

Já o escalonador DRR é adequado para redes convergentes, sendo que somente no primeiro cenário analisado, onde há apenas tráfego de dados, é que ele teve seu desempenho sutilmente inferior ao escalonador FIFO. No entanto, ele comporta um número aceitável de usuários de voz no sistema, sendo que o mesmo chega a entrar em estabilidade. Contudo, no que tange ao tráfego de vídeo, ainda há algumas restrições a serem analisadas. No geral, seu desempenho é melhor que o escalonador FIFO, admitindo um maior número de usuários no sistema.

No tocante ao escalonador DRR que calcula a quota de cada tráfego com base na banda efetiva proposta em [62], nos cenários analisados, conseguiu atender ao quesito de QoS no que se refere a perda de pacotes, sendo que nenhum pacote foi perdido. Contudo, o tempo de sistema foi elevado, sendo este superior a 500 s para o tráfego de dados. Isso não ocorre usando os escalonadores FIFO e DRR, que possuem tempo de sistemas, no pior caso, de 160 s para o tráfego de vídeo. Deste modo, uma alternativa que controle ambos os parâmetros de QoS ainda deve ser investigada.

Para propor escalonador conveniente para redes de sensores corporais, notou-se que na literatura examinada, a maioria baseia em TDM e poucos citam o escalonador *polling*. Desta forma, nesta tese, investigou-se a conveniência da utilização do escalonador *polling* para redes de sensores corporais. Para isso, desenvolveu-se uma plataforma de simulação contemplando diversos modelos de sensores, buffers e um servidor cíclico que emula o escalonador *polling*. Os modelos de sensores desenvolvidos são inéditos na literatura.

A rede de sensores simulada foi em pequena escala, contemplando somente sete sensores. Entretanto, alguns resultados obtidos poderão ser importantes para futuro desenvolvimento dessa rede. Uma das conclusões é que três posições de buffer são suficientes para atender aos requisitos de perda de pacotes e tempo de espera. Além disso, pode-se concluir que o escalonador *polling* pode ser uma boa alternativa para coletar dados dos sensores sobre ou subcutâneos implantados no corpo humano.

6.1 TRABALHOS FUTUROS

Durante a elaboração e após a conclusão deste trabalho, foram verificados vários pontos que podem ser melhorados e complementados. Diante disso, seguem algumas sugestões para realização de trabalhos futuros.

Implementar modelos de fonte auto-similares e inseri-las nos escalonadores FIFO, DRR e DRR com distribuição de quota usando a banda efetiva proposta em [62], visando comparar as diferenças que os modelos de tráfegos acarretam nos equipamentos de interconexão e sistemas de redes.

Desenvolver a função para cálculo de banda efetiva para fontes auto-similares, de modo que este parâmetro seja usado para a definição do quantum de cada fila.

Verificar se usando o cálculo da banda efetiva proposta em [50], o qual atende ao tráfego On/Off, gera diferenças significativas no modo de escalonamento, comparando com os resultados obtidos com o escalonador desenvolvido e proposto neste trabalho.

Implementar outros algoritmos de escalonamento e confrontar os resultados obtidos com os valores dos escalonadores FIFO, DRR e o novo algoritmo de escalonamento proposto.

No tocante as Redes Corporais também foram identificadas melhorias a serem contempladas, a citar:

- implementar novos mecanismos de Controle de Acesso ao Meio (MAC), como o CSMA/CA, TDMA, para confrontar os valores obtidos com os apresentados pelo mecanismo polling;
- conhecer diferentes parâmetros para a geração das fontes, bem como o desenvolvimento de novas fontes que simulem de forma fiel o comportamento de uma RSC. Isso se deve ao fato de que se os sensores forem implantados no corpo humano, é preciso verificar o quanto a corrente sanguínea influencia na transmissão das informações;

- conhecer os custos energéticos das funções desempenhadas pelos sensores é outro fator que deve ser analisado. Desta forma será possível inserir estes parâmetros nos simuladores para que gerem pacotes enquanto existir energia na bateria.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] 3GPP2 WG5 Evaluation Ad hoc. “1xEV-DV Evaluation methodology- addendum (V6)”, July 2001.
- [2] AALSALEM, M. Y; IFTIKHAR, Mohsin; TAHERI, Javid; ZOMAYA, Albert Y.. On the Provisioning of Guaranteed QoS in Wireless Sensor Networks through Limited Service Polling Models. In Proceedings of 5th IFIP International Conference on Wireless and Optical Communications Networks, 2008 (WOCN '08). 5-7 May 2008. pp. 1 – 7.
- [3] AKYILDIZ, Ian F.; MELODIA, Timmaso; CHOWDHURY, Kaushik R. A survey on wireless multimedia sensor networks. Computer Networks (Elsevier), Vol. 51, pp. 921-960, 2007.
- [4] ALI, Khaled A.; SARKER, Jahangir H.; MOUFTAH, Hussein T. QoS-based MAC protocol for medical wireless body area sensor networks. In Proceedings of the IEEE Symposium on Computers and Communications (ISCC'2010), 2010. pp. 216-221.
- [5] ALI, Khaled A.; SARKER, Jahangir H.; MOUFTAH, Hussein T.. Urgency-Based MAC Protocol for Wireless Sensor Body Area Networks. Proceedings of the IEEE International Conference on Communications Workshops (ICC), 2010. 23-27 May 2010. pp. 1-6.
- [6] ALLAHYAR, Ehsan; MOVAGHAR, Ali. Weighted Start-time Fair Queuing: A New Start-time Fair Queuing For Elastic Traffic. In Proceeding SpringSim '09 Proceedings of the 2009 Spring Simulation Multiconference. Society for Computer Simulation International, San Diego, CA, USA.
- [7] ALTINTAS, Onur; ATSUMI, Yukio; YOSHIDA, Teruaki. A Note on Fair Queueing and Best-Effort Service in the Internet. Ultra-high Speed Network and Computer Technology Laboratories (UNCL), Toranomom 5-2-6, Minato-ku, Tokyo 105-0001, Japan, 1999.

- [8] AMERICANAS. 2010. Disponível em: <http://www.americanas.com.br/AcomProd/6872/285179>. Acesso em: 20/09/ 2010
- [9] ARAS, Caglan M; KUROSE, James F; REEVES, Douglas S; SCHULZRINNE, Henning. Real-Time Communications in Packet-Switched Networks. In: Proceedings of the IEEE, vol. 82, no. 1, pp. 122-139, Jan, 1994.
- [10] BACK, Doo-sung; PYUN, Kihyun; LEE, Seung-Min; CHO, Junhee; KIM, Namsu. A Hierarchical Deficit Round-Robin Scheduling Algorithm for a High Level of Fair Service. International Symposium on Information Technology Convergence, pages 115-119, Joenju, 2007.
- [11] BALLESTEROS, Luis Guillermo Martínez. Towards a Methodology to Define Service Level Agreements in a Convergent Network Scenario. ANDESCON. Researcher Engineering School, University Sergio Arboleda, Bogota, Colombia, 2010.
- [12] BARBOSA, Talles Marcelo Gonçalves de Andrade. Uma Arquitetura de Redes Sensores do Corpo Humano. Tese (Doutorado em Engenharia Elétrica) – Faculdade de Tecnologia, Universidade de Brasília, Brasília, 2008.
- [13] BI, Qi; CHENG, Pi-Chun; YANG, Yang; ZHANG, Qinqing. An Analysis of VoIP Service Using 1 EV-DO Revision A System. IEEE Journal On Selected Areas In Communications, Vol. 24, No. 01, pages 36-45. Whippany, NJ, 2006.
- [14] BLAKE, S.; BLACK, D.; CARSON, M.; DAVIES, E.; WANG, Z.; WEISS, W.. An Architecture for Differentiated Services. RFC-2475. Network Working Group. Dezembro, 1998.
- [15] BODAMER, Stefan; CHARZINSKI, Joachim. Evaluation of Effective Bandwidth Schemes for Self-Similar Traffic. Proceedings of the 13th ITC Specialist Seminar on IP Measurement, Modeling and Management, Monterey, CA, September 2000, pp. 21-1-21-10.
- [16] BONFIM, Tatiane Regina. Construção de um Simulador de Redes ATM baseado em Redes de Petri. 2001. 97 f. Dissertação (Mestrado em Engenharia Elétrica) – Dissertação, Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação, Campinas, 2001.
- [17] BORGES, Luis M.; VELEZ, Fernando J.; LEBRES, Antonio S.. Performance Evaluation of the Schedule Channel Polling MAC Protocol applied to Health Monitoring in the Context of IEEE 802.15.4. Proceedings of the 11th European Wireless Conference - Sustainable Wireless Technologies (European Wireless), 2011. pp. 94-101.

- [18] BROWNFIELD, Michael I; NELSON, Theresa; MIDKIFF, Scott; DAVIS IV, Nathaniel J. Wireless Sensor Network Radio Power Management and Simulation Models. In: The Open Electronic Engineering Journal. 2010.
- [19] BUNN, Alexandre. AVALIAÇÃO DO SERVIÇO DE QOS NO FREEBSD. 2004. Monografia (Graduação em Sistemas de Informação) - Universidade Federal de Santa Catarina. Florianópolis, 2004.
- [20] CALLEGATI, Franco; TAMIL, Lakshman. On Optical Burst Switching and Self-Similar Traffic. IEEE Communications Letters, Vol. 4, N°. 3, March 2000.
- [21] CASTILHO, Marzon Pereira de. Projeto de Protocolos considerando as características não lineares das baterias em Redes de Sensores Sem Fio. Dissertação (Mestrado em Ciência da Computação) – Pontifícia Universidade Católica de Minas Gerais, Belo Horizonte, 2009.
- [22] CAVALCANTI, D.; SCHMITT, R.; SOOMRO, A. Achieving Energy Efficiency and QoS for Low-Rate Applications with 802.11e. In: Wireless Communications and Networking Conference, Kowloon, 2007.
- [23] CHAKRABARTY, Krishnendu; IYENGAR, Sundararaja S.. Scalable infrastructure for distributed sensor networks. Springer-Verlag, 2005.
- [24] CHANDRASEKARAN, Balakrishnan. Survey of Network Traffic Models. CSE567M: Computer Systems Analysis. Washington of St. Louis, 2006. Disponível em: http://www1.cse.wustl.edu/~jain/cse567-06/traffic_models3.htm. Acesso em: 11 de agosto de 2011.
- [25] CHEVALIER, Philippe B; WEIN, Lawrence M. Scheduling Networks of Queues: Heavy Traffic Analysis of a Multistation Closed Network. Massachusetts Institute of Technology, Cambridge, Massachusetts, 1990. Disponível em: <http://faculty-gsb.stanford.edu/wein/personal/SchedulingNetwork.pdf>, acesso dia 23-03-2011.
- [26] CHOI, Jin Seek; BAE Bum Sik; LEE, Hyeong Ho; LEE Hyeong Sub. Round-Robin Scheduling Algorithm with Multiple Distributed Windows. In: Information Networking: Wireless Communications Technologies and Network Applications Lecture Notes in Computer Science, 2002, V. 2344/2002, p. 814-819.
- [27] CHOI, Min Seok; YAQUIONG, Wang ; KIM, Kyung Tae; YONG, Suk Jung; YOUN, Hee Yong. A New Polling Protocol with Efficient Cluster Creation for WSN. IEEE/ACM

- International Conference on Green Computing and Communications & 2010 IEEE/ACM International Conference on Cyber, Physical and Social Computing, 2010. pp 357-362.
- [28] Cisco IOS Quality of Service Solutions Configuration Guide. Congestion Management Overview. Disponível em: <http://www.cisco.com/en/US/docs/ios/12_2/qos/configuration/guide/qcfconmg.pdf>. Acesso em: 31 maio 2011.
- [29] Cisco_Corporation. Chapter 49 - Quality of Service Networking. Internetworking technology handbook, Cisco_Corporation, Editor, Cisco Systems. p. 32., 2007.
- [30] CROSSBOW. MicaZ Datasheet. 2008. Disponível em: http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICAz_Datasheet.pdf Acesso em: 21/09/ 2010.
- [31] CROWCROFT, Jon; HAND, Steven; MORTIER, Richard; ROSCOE, Timothy; WARFIELD, Andrew. Plutarch: An Argument for Network Pluralism. Proceedings of the Workshop on future Directions in Network Architecture (FNDA) at ACM SIGCOMM, Karlsruhe, Germany, August 27, 2003.
- [32] DAS, Subir; ZUNIGA, Juan Carlos; CYPHER, David; CHAN, Anthony. "IEEE 802.21." December, 2006. Disponível em: <http://www.ieee802.org/21/>. Acesso em: 11/12/2011.
- [33] DEMERS, Alan; KESHAV, Srinivasan; SHENKER, Scott. "Analysis and simulation of a fair queueing algorithm," in ACM SIGCOMM Computer Communication Review, vol. 19, no. 4, Sept. 1989, pp. 1-12.
- [34]DEUTSCHE BANK RESEARCH. The Dawn of Technological Convergence. Economics magazine, ISSN 1619-3245, No. 56. IT, telecoms & New Media. Frankfurt, Germany, 2006.
- [35]DHARWADKAR, Shri N.; MASOOD, Nabegha. Next Generation Network. Consumer Electronics 2007. ISCE 2007. IEEE International Symposium on 20-23 June 2007. pp. 1-4, Irving, TX, 2007.
- [36] DIOURI, Idriss; GEORGES, Jean-Philipp; RONDEAU, Eric. Accommodation of delays for networked control systems using classification of service. In Proceedings of the 2007 IEEE International Conference on Networking Sensing and Control (2007). London, 15-17 April 2007, pp: 410-415.

- [37] ELLIS, Juanita; PURSELL, Charles; RHAMAN, Joy. Voice, Video, and Data Network Convergence: Architecture and Design, From VoIP to Wireless. Elsevier Science: USA, 2003.
- [38] ERRAMILI, Ashok; VEITCH, Darryl; ROUGHAN, Matthew; WILLINGER, Walter. Self-Similar Traffic and Networks Dynamics. Proceedings of the IEEE, Vol. 90, No. 5, May 2001.
- [39] FALL, Kevin. Scheduling Best-Effort and Guaranteed Connections. 1999. EE122 / lec27. <http://www.cs.berkeley.edu/~kfall/EE122/lec27/tsld001.htm>. Acesso em: 25 mar. 2006.
- [40] FANG, Gengfa; DUTKIEWICZ, Eryk. BodyMAC: Energy efficient TDMA-based MAC protocol for Wireless Body Area Networks. Proceedings of the 9th International Symposium on Communications and Information Technology (ISCIT 2009), 2009. pp. 1455-1459
- [41] FENG, Benjamin Z. M; HUAN, Chang Cheng. FISTE: A Black Box Approach for End-to-End QoS Management. ACM Transactions on Modeling and Computer Simulation, Vol. 19, N°. 4, Article 16, Publication date: October 2009.
- [42] FERNANDO, Amila; WILLIAMS, Darren; FEKETE, Alan; KUMMERFELD, Bob. Dynamic Network Service Installation in an Active Network. Submission for Special Issue of the Computer Networks Journal on Active Networks and Services. Vol. 36, pp. 35 - 48, 2001.
- [43] FIDALGO, Joseane. F; KAMIENSKI, Carlos Alberto; FARIAS, Thiago Souto Maior Cordeiro de; DIAS, Kelvin Lopes; ALMEIDA, Fábio Guerra; SADOK, Djamel Fawzi; KELNER, Judith. QNet – Um Simulador Gráfico de Tráfego IP para redes convergentes. In. 22º Simpósio Brasileiro de Redes de Computadores, 2004, Gramado. Anais. Gramado: Universidade Federal do Rio Grande do Sul, 2004. Disponível em: http://www.di.ufpe.br/~cak/publications/sbrc2004_qnet.pdf. Acesso em: 09/08/2011.
- [44] FISCHER, Martin J.; HARRIS, Carl. “A Method for Analyzing Congestion in Pareto and Related Queues” The Telecommunications Review, pp. 15-27, 1999.
- [45] FONSECA, Nelson L. S., MAYOR, Gilberto S., NETWORKO, Cesar A. V., On the equivalent bandwidth of self-similar sources. ACM Transactions on Modeling and Computer Simulation (TOMACS), v.10 n.2, p.104-124, April 2000.
- [46] FRANTTI, Tapio; JUTILA, Mirjami. Embedded fuzzy expert system for Adaptive Weighted Fair Queueing. Expert Systems with Applications: An International Journal - Volume 36 Issue 8, October, 2009. Pergamon Press, Inc. Tarrytown, NY, USA.

- [47] FROST, Victor. S; MELAMED, Benjamin. Traffic Modeling for Telecommunications Networks. IEEE Communications Magazine, University of Kansas, March 1994.
- [48] FUROUZAN, Behrouz A. Data Communications and Networking. 4th Ed. The McGraw-Hill Companies. Nova York. 2007.
- [49] GRAVINA, Raffaele; GUERRIERI, Antonio. FORTINO, Antonio. Development of Body Sensor Networks Applications using SPINE. In: IEEE International Conference on Systems, Man and Cybernetics. Singapura, 2008.
- [50] GUÉRIN, R.; AHMADI, H.; NAGHSHINEH, M.. “Equivalent capacity and its application to bandwidth allocation in high-speed networks”. IEEE J. Select Areas Commun, vol. 9, pp.968-981, 1991.
- [51] HALKES, G.P.; LANGENDOEN, K.G. Experimental Evaluation of Simulation Abstractions for Wireless Sensor Network MAC Protocols. In: EURASIP Journal on Wireless Communications and Networking. V. 2010. Hindawi Publishing Corporation.
- [52] HANRAHAN, Hu. Network Convergence: Services, Applications, Transport, and Operations Support. John Wiley & Sons: South Bend, IN, U.S.A., 2007.
- [53] HASSAN, Hassan; GARCIA, Jean-Marie; BRUN, Oliver. Generic Modeling of Multimedia Traffic Sources. Laboratory for Analysis and Architecture of Systems. Toulouse, France, 2005.
- [54] HENRIQUES, Felipe da Rocha; LOSOVIO, Lisandro; RUBINSTEIN, Marcelo Gonçalves. Algoritmos para Aumentar o Tempo de Vida de Redes de Sensores Sem Fio Utilizando Inovação. In: XXVII Simpósio Brasileiro de Telecomunicações. Blumenau – SC, 2009.
- [55] HERSCOVICI, Naftali; CHRISTODOULOU, Christos; LAPPAS, V.; PRASSINOS, G.; BAKER, A.; MAGNUSS, R. Wireless Sensor Motes for Small Satellite Applications. IEEE Antennas and Propagation Magazine, Vol. 48, No. 5, October 2006. pp. 175-179.
- [56] HWANG, Fernando. Análise dos Efeitos Gerados pelo Comportamento das Aplicações e pelo Perfil das Redes na Característica Auto-Similar do Tráfego Internet. 2004. 167 f. Dissertação (Mestrado) - Departamento de Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, 2004.
- [57] HYUN-HWA, Seo; BYUNG-HAN, Ryu; CHOONG-HO, Cho; HYONG-WOO, Lee. “Traffic Characteristics Based Performance Analysis Model for Efficient Random Access in OFDMA-

- PHY System”, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, Volume 3510/2005.
- [58] ILYAS, Mohammad, MAHGOUB, Imad. The Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems. Washington, D.C: CRC Press, 2005.
- [59] JONIT, Norhayati Md; BABA, Mohd Dani. First In First Out (FIFO) and Deficit Round Robin (DRR) Scheduling Analysis in WiMAX Network. IEEE Control and System Graduate Research Colloquium, Malaysia, 2011.
- [60] JOVANOVIĆ, Emil; MILENKOVIC, Aleksandar; OTTO, Chris; GROEN, Piet C. A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation Journal of NeuroEngineering and Rehabilitation, Vol. 2, No. 1, 1 March 2005. doi:10.1186/1743-0003-2-6 Key: citeulike:767908. Disponível em: <http://www.jneuroengrehab.com/content/2/1/6>.
- [61] KELLY, F. P. Notes on Effective Bandwidths. In Stochastic Network: Theory and Applications, ser. Royal Statistical Society Lecture Notes. Oxford University Press, 1996, vol. 4, pp. 141–168.
- [62] KESIDIS, G.; WALRAND J.; CHANG, C.-S.. Effective bandwidths for multiclass Markov fluids and other ATM sources. IEEE Trans. Network, vol.1, n° 4, pp. 424-28, Aug. 1993.
- [63] KIM, Sunggon; NAM, Seung Yeob; SUNG, Dan Keun. Effective bandwidth for a single server queueing system with fractional Brownian input. Performance Evaluation, v.61 n.2-3, p.203-223, July 2005.
- [64] KOCHER, Ana Cristina Barreiras. Controle de admissão de chamadas, reserva de recursos e escalonamento para provisão de QoS em redes GPRS. Dissertação (Mestrado em Engenharia Elétrica e Informática Industrial) - Centro Federal de Educação Tecnológica do Paraná, Curitiba, 2003.
- [65] KUROSE, James F; ROSS, Keith W. “Redes de computadores e a internet: uma abordagem top-down”. 3 ed. São Paulo: Pearson Addison Wesley, 2006.
- [66] KUTTY, Shajahan; LAXMINARAYAN, J. A.. Towards energy efficient protocols for wireless body area networks. Proceedings of the International Conference on Industrial and Information Systems (ICIIS), 2010. pp. 31-34.

- [67] LATRÉ, Benoit; BRAEM, Bart; MOERMAN, Ingrid; Blondia, Chris; DEMEESTER, Piet. A Survey on Wireless Body Area Networks. *Proceedings of the Wireless Networks*. Volume 17, Issue 1. January, 2011. Kluwer Academic Publishers Hingham, MA, USA.
- [68] LAYERLE, David; KWASINSKI, Andres. A power efficient pulsed MAC protocol for Body Area Networks. *Proceedings of the IEEE 22nd International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, 2011.
- [69] LELAND, Will E.; TAQQU, Murad S.; WILLINGER, Walter; WILSON, Daniel. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Transactions Networking*, Vol. 2, n. 1, pp. 1–15, Feb. 1994.
- [70] LI, Changle; LI, Huan-Bang; KOHNO, Ryuji. Performance Evaluation of IEEE 802.15.4 for Wireless Body Area Network (WBAN). *Proceedings of the IEEE International Conference on Communications Workshops (2009)*, 2009. 14-18 June 2009.
- [71] LI, Huaming; TAN, JinDong. An Ultra-low-power Medium Access Control Protocol for Body Sensor Networks. In: *Proceedings of 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*. Shanghai, 2005.
- [72] LI, Huaming; TAN, JinDong. Heartbeat-Driven Medium-Access Control for Body Sensor Networks. *Proceedings of the IEEE Transactions on Information Technology in Biomedicine*, Vol. 14, No. 1, January 2010. pp. 25-30.
- [73] LIU, Bin; YAN, Zhisheng; CHEN, Chang Wen. CA-MAC: A Hybrid context-aware MAC protocol for wireless body area networks. *Proceedings of the 13th IEEE International Conference on e-Health Networking Applications and Services (Healthcom)*, 2011. pp. 213-216.
- [74] LIU, Heng; JU, Fang. A Novel Packet Scheduling Algorithm based on Self-Similar Traffic in WSN. In *Proceedings International Conference on Wireless Sensor Network*, 2010. 15-17, Nov. 2010. pp: 312 – 316.
- [75] LIU, Nelson X.; BARAS, John S.. “Statistical Modeling and Performance Analysis of Multi-Scale Traffic” *IEEE INFOCOM* 2003.
- [76] LIU, Qianlin; ZHAO, Dongfeng. Priority Polling Strategy for Wireless Sensor Networks. *1st International Conference on Information Science and Engineering (ICISE)*, 2009. 26-28 Dec. 2009. pp. 4062 – 4065.

- [77] LOUREIRO, Antonio A. F.; NOGUEIRA, José M. S.; RUIZ, Linnyer B.; MINI, Raquel, A. F.; NAKAMURA, Eduardo F.; FIGUEIREDO, Carlos M. S. Redes de Sensores Sem Fios. In: XXI Simpósio Brasileiro de Redes de Computadores; 2003, Natal, Rio Grande do Norte.
- [78] MA, Junchao; LOU, Wei; WU, Yanwei; LI, Xiang-Yang. Energy Efficient TDMA Sleep Scheduling in Wireless Sensor Networks. In: IEEE INFOCOM 2009. Rio de Janeiro, 2009.
- [79] MACEDO, Daniel F.; SANTOS, Aldri Luiz dos; PUJOLLE, Guy. From TCP/IP to Convergent Networks: Challenges and Taxonomy. In: Proceedings of IEEE Communications Surveys & Tutorials, Vol. 10, N°. 4, Fourth Quarter, 2008. PP. 40-55.
- [80] MAGALHÃES, Maurício F.; CARDOSO, Eleri. “Qualidade de Serviço na Internet”. Versão Draft. Unicamp, 1999. Documento disponível em: <ftp://ftp.dca.fee.unicamp.br/pub/docs/elери/apostilas/>
- [81] MARINKOVIC, Stevan; SPAGNOL, Christian; POPOVICI, Emanuel. Energy-Efficient TDMA-Based MAC Protocol for Wireless Body Area Networks. Proceedings of the Third International Conference on Sensor Technologies and Applications (SENSORCOMM '09) 2009. 18-23 June 2009. pp. 604 – 609.
- [82] MARQUES, Leandro Bento Sena. “Estudo de Desempenho do Sistema 3G 1xEV-DO Através de Modelos Reais de Tráfego”. 2005. Dissertação (Mestrado em Engenharia Elétrica e de Computação) - Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. Campinas, 2005.
- [83] MARTINEZ, Daniel Alonso. A Verification of Selected Properties of Telecommunications Traffic Generated By Opnet Simulator. Monografia (Trabalho de projeto de intercâmbio), University of Ljubljana, Faculty of Electrical Engineering, 2011.
- [84] MARTÍNEZ, Raúl; ALFARO, Francisco J.; SÁNCHEZ, José L; CLAVER, José M. Hardware Implementation Study of the SCFQ-CA and DRR-CA Scheduling Algorithms. Euro-Par 2009 Parallel Processing, Lecture Notes in Computer Science, 2009. PB - Springer Berlin / Heidelberg, SP - 1089, EP - 1100, VL - 5704.
- [85] MATA, Renê Souza da. Dimensionamento de enlaces em redes com integração de serviços. 2002. Dissertação (Mestrado em Engenharia Elétrica e de Computação) - Universidade Estadual de Campinas. Campinas, 2002.

- [86] MATSUOT, Takahiro; HASEGAWAT, Go; MURATAT, Masayuki; MIYAHARAT, Hideo. Comparisons of Packet Scheduling Algorithms for Fair Service among Connections. International Workshop – IWS, pages 137-144, Osaka, Japan, 1999.
- [87] MEDHI, Deepankar; RAMASAMY, Karthikeyan. Network routing: algorithms, protocols, and architectures. Morgan Kaufmann: San Francisco, 2007.
- [88] MENDES FILHO, Jorge Roberto. Uma extensão ao modelo de tráfego auto-similar com aplicações em controle de admissão e tarifação em redes ATM. Dissertação (Mestrado em Engenharia Elétrica) – Dissertação. Universidade Federal do Rio de Janeiro – Rio de Janeiro, 1998.
- [89] MESSIER, Geoffrey G.; FINVERS, Ivars G. Traffic models for medical wireless sensor networks. IEEE Communications Letters. Vol. 11, No. 1, pp. 13-15, January 2007.
- [90] MUHAMMED, Abdullah; MOHD, KyairulNiza Saleh; ABDULLAH, Azizol. Performance Comparison of Voice Packet Sizes in the FIFO Adversarial Queuing and FIFO M/M/1 Model. IJCSNS International Journal of Computer Science and Network Security, Vol. 8, No.1, January 2008.
- [91] NORROS, Ilkka. “A storage model with self-similar input” Queuing Systems, Vol. 16, Numbers 3-4, pp. 387-396, 1994.
- [92] NOURIZADEH, Shahram; SONG, Y. Q; THOMESSE, J.P. An Adaptive Hierarchical Routing Protocol for Wireless Ad-hoc Sensor Networks. In: Third International Conference on Next Generation Mobile Applications, Services and Technologies (IEEE NGMAST’09). Cardif, Wales. 2009.
- [93] OLIVEIRA, J. C., BONATTI, I. S., PERES, P. L. D. and BUDRI, A. K.. Cell level performance approach for link dimensioning of ATM networks. Proceedings of ITS 98, 1998, vol.1, pp. 189-194, São Paulo, SP, Brazil.
- [94] OLIVEIRA, Sandro Silva de; MOTOYAMA, Shusaburo. Applications Oriented Medium Access Control Protocols for Wireless Sensor Networks. Proceedings of the IEEE Latin America Transactions, v. 7, Issue 5, 2009. pp. 586-593.
- [95] OTAL, B.; ALONSO, Luis Gonzaga; VERIKOUKIS, Ch. Novel QoS Scheduling and Energy-saving MAC protocol for Body Sensor Networks Optimization, In Proceedings of the Third International Conference on Body Area Networks (BodyNets08), Tempe, Arizona, 2008.

- [96] OTAL, B.; ALONSO, Luis Gonzaga; VERIKOUKIS, Ch. Towards Energy Saving Wireless Body Sensor Networks in Health Care Systems. In Proceedings of IEEE International Conference on Communications (ICC 2010), Second International Workshop on Medical Applications Networking (MAN 2010), Capetown, África dos Sul, 2010.
- [97] OTERO, Daniel Cavas. Alternativas para Diferenciação de Serviços em Redes Locais sem Fio. Dissertação (Mestrado em Engenharia de Sistemas e Computação) – Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2004.
- [98] OTTO, Chris; MILENKOVIĆ, Aleksandar; SANDERS, Corey; JOVANOVIĆ, Emil. System Architecture of a Wireless Body Area Sensor Network for ubiquitous health monitoring. Journal of Mobile Multimedia, Vol. 1, No. 4, pp. 307-326, 2006.
- [99] PAL, Arindarjit; SINGH, Jyoti Prakash; DUTTA, Paramartha. A Study on the Effect of Traffic Patterns in Mobile Ad hoc Networks. ACC, Part I, CCIS 190, pp. 83-90, Springer-Verlag Berlin Heidelberg, 2011.
- [100] PAPAPANAGIOTOU, Ioannis; VARDAKAS, John. S; PASCHOS, Georgios. S; LOGOTHETIS, Michael.D; KOTSOPOULOS, Stravos.A. “Performance Evaluation of IEEE 802.11e based on ON-OFF Traffic Model” ACM International Conference Proceeding Series; Vol. 329. Proceedings of the 3rd international conference on Mobile multimedia communications, 2007.
- [101] PAZETO, Tatiana Annoni; MOTOYAMA, Shusaburo. “Escalonamento de Tráfego Usando a Banda Efetiva de Kesidis para Garantir QoS em Redes Convergentes”. In: 5th International Information and Telecommunication Technologies Symposium (I2TS 2006), December 06-08, 2006. Cuiabá, MT, Brazil.
- [102] PAZETO, Tatiana Annoni; MOTOYAMA, Shusaburo. QoS Guaranteed Traffic Scheduling in Convergent Networks Using Effective Bandwidth. In: International Workshop on Telecommunications - IWT 2007, 2007, Santa Rita de Sapucaí-MG. International Workshop on Telecommunications, 2007. v. 1. p. 142-147.
- [103] PORTNOI, Marcos. Estudo de Características de Fontes de Tráfego para Redes de Computadores Multi-Serviço. Trabalho de conclusão de curso (Graduação em Engenharia Elétrica) - Universidade de Salvador, Salvador, 2003.

- [104] REFATTI, Luis Fernando; PAZETO, Tatiana Annoni. Análise e Impacto das Fontes Para Redes de Sensores para o Corpo Humano. In: V Congresso Sul Brasileiro de Computação, 2010, Criciúma - SC.
- [105] ROLLS, David A.; MICHAILIDIS, George; HERNANDEZ-CAMPOS, Félix. “Queueing analysis of network traffic: methodology and visualization tools”. Computer Networks, Vol. 48, N°. 3, pp. 447-473, June 2005.
- [106] RONG, Lu; RUIMIN, Lu; ZHIGANG, Cao, Application of cross-layer design in satellite communication networks. Wireless Communications & Signal Processing, 2009. WCSP 2009. Conferência Internacional sobre , vol., não., pp.1-5, 13-15 novembro 2009.
- [107] RUIZ, Linnyer B.; CORREIA, Luiz H.; VIEIRA, Luiz F.; MACEDO, Daniel F.; NAKAMURA, Eduardo F.; FIGUEIREDO, Carlos M. F.; VIEIRA, Marcos Augusto M.; MECHELANE, Eduardo H.; CAMARA, Daniel; LOUREIRO, Antonio A. F.; NOGUEIRA, José Marcos S.; SILVA JUNIOR, Diógenes. Arquitetura de redes de sensores sem fio. Pág. 167–218. Capítulo 4 do livro texto de mini-cursos do XXII Simpósio Brasileiro de Redes de Computadores (SBRC). Gramado, RS, Brasil, 2004. ISBN 85-88442-81-7
- [108] SALVADOR, Paulo; NOGUEIRA, António; VALADAS, Rui. Markovian Models for Medical Signals on Wireless Sensor Networks. ICC 2009 Workshop - First International Workshop on Medical Applications Networking (MAN 2009), 14 June, 2009, Dresden, Germany. ISBN: 978-1-4244-3437-4. doi:10.1109/ICCW.2009.5208090
- [109] SEMERIA, Chuck. Supporting Differentiated Service Classes: Queue Scheduling Disciplines - White Paper, tech. report, Juniper Networks, 2001.
- [110] SENE JR, I. G.; BARBOSA, T. M. G. A.; ROCHA, A. F.; NASCIMENTO, F. A. O.; CARVALHO, H. S. Monitoração de Temperatura Corporal Baseada em uma Rede de Sensores Sem Fios”, XVIII Congresso Brasileiro de Informática em Saúde (CBIS), 2006.
- [111] SHREEDHAR, M; VARGHESE, George. “Efficient fair queuing using Deficit Round Robin”. IEEE/ACM Transactions on Networking, v. 4, n. 3, p 375-385, June 1996.
- [112] SICHITIU, Mihail L. Cross-Layer Scheduling for Power Efficiency in Wireless Sensor Networks. In: Proceedings of INFOCOM 2004, Hong Kong: 2004.
- [113] SILVA, Adailton J. S. “Qualidade de serviço em VoIP – parte I”. Boletim bimestral sobre tecnologia de redes produzido e publicado pela RNP – Rede Nacional de Ensino e Pesquisa. 12

- de maio de 2000, Vol 4, N. 3. ISSN 1518-5974. Disponível em: http://www.rnp.br/newsgen/0005/qos_voip1.html. Acesso em: 14 abr. 2006.
- [114] SILVA, Dinailton José da. “Análise de qualidade de serviço em redes corporativas”. 2004. Dissertação (Mestrado em Engenharia Elétrica e de Computação) – Universidade Estadual de Campinas. Campinas, 2004.
- [115] SILVA, Renato Moraes; PAZETO, Tatiana Annoni. Proposals of On/Off Sources and their effects on the DRR and FIFO schedulers, indicating the most influential parameter. In: International Conference on Computer Communications and Network (CCN-10), realizado de 12 a 14 de julho de 2010 em Orlando, Florida, USA. Esta conferência faez parte do MULTICONF-10. ISBN: 978-1-60651-018-6.
- [116] SLIMANE, Jamila Ben; SONG, Ye-Qiong; KOUBAA, Anis. A Prioritized Multi-Channel Multi-Time slot MAC Protocol For Large-Scale Wireless Sensor Networks. In: The First International Conference on Communications and Networking (IEEE ComNet’09). Hammamet, 2009.
- [117] SOARES, Luiz Fernando G; LEMOS, Guido; COLCHER, Sérgio. Redes de Computadores: das LANs MANs e WANs as Redes ATM. 2ª edição. Rio de Janeiro: Campus, 1995.
- [118] STILIADIS, Dimitiros; VARMA, Anujan. Latency-rate servers: a general model for analysis of traffic scheduling algorithms. IEEE/ACM Transactions on Networking, Vol. 6, N. 5, OCTOBER 1998.
- [119] TANG, Shensheng. An analytical traffic flow model for cluster-based wireless sensor networks. In 1st International Symposium on Wireless Pervasive Computing, 2006.
- [120] TEIXEIRA, Ingrid. Roteamento com balanceamento de consumo de energia para Rede de Sensores Sem Fio. Tese (Mestrado em Ciência da Computação) – Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2005.
- [121] TEXAS INSTRUMENTS. Ultralow Power Transceiver and Smart Sensor Interface Platform. 2009. Richard McPartland (enviado por email) <richard.mcpartland@toumaz.com >. 18/05/2010.
- [122] TIMMONS, Nicholas F.; SCANLON, William G. Analysis of the Performance of IEEE 802.15.4 for Medical Sensor Body Area Networking. In Proceedings of IEEE Conference on Sensor, Ad Hoc Communications and Networks (SECOM 2004). 2004. pp. 16-24.

- [123] TSELISHCHEV, Yuriy. Designing a Medium Access Control protocol for Body Area Networks. Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2011. 20-24 June 2011. pp. 1-3.
- [124] TSENG, Hung-Wei; YANG, Shih-Hsien; CHUANG, Po-Yu; KUANG WU, Eric Hsiao; CHEN, Gen-Huey. An Energy Consumption Analytic Model for a Wireless Sensor MAC Protocol. In Proceedings of the IEEE Vehicular Technology Conference (VTC'2004), September 2004, pp. 4533-4537.
- [125] TUREK, Melanie. Voice and video over IP: leveraging network convergence for collaboration. Issue Paper, Nemertes Research, 2006.
- [126] ULLAH, S.; KHAN, P.; ULLAH, N.; SALEEM, S.; HIGGINS, H.; KWAK, K. S. A Review of Wireless Body Area Networks for Medical Applications. In: Int'l J. of Communications, Network and System Sciences. N.8, 2009.
- [127] VERÍSSIMO, Fernando Carlos Azeredo. “Propostas e avaliações de protocolos de acesso alternativos ao padrão IEEE 802.11e”. Dissertação (Mestrado em Engenharia de Sistemas e de Computação). Universidade Federal do Rio de Janeiro. Rio de Janeiro, 2005.
- [128] VILELA, João; LIMA, Luísa. Mecanismos de escalonamento de pacotes. 2004. Faculdade de Ciências da Universidade do Porto. Porto, 2004. Disponível em: <http://www.dcc.fc.up.pt/~pbrandao/aulas/0405/TAR/relatorios/0304-t2-scheduling.pdf>. Acesso em: 14 abr. 2006.
- [129] VIRONE, Gilles; WOOD, Anthony D.; SELAVO, Leo; CAO Quihua; FANG, Lei; DOAN, Thao; HE, Zhimin; STOLERU, Radu; LIN, Shu; STANKOVIC, John. An Advanced Wireless Sensor Network for Health Monitoring. In Proceedings of the 1st Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare (D2H2), Arlington, VA, April 2--4, 2006.
- [130] WILLINGER, Walter; PAXSON, Vern. Where Mathematics Meets the Internet. In: Notices of the ACM, Vol 45(8), pp. 961-970, 1998.
- [131] WU, Eirc Hsiao-Kuang; LAI, Hsu-Te; TSAI, Meng-Feng; CHOU, Cheng-Fu. Low Latency and Efficient Packet Scheduling for Streaming Applications. ICON – 12th IEEE International Conference on Network, pages 1413-1421, Amsterdam 2004.

- [132] XIUMING, Zhu; SONG, Han; PEI-CHI, Huang; MOK, Aloysius K.; DEJI, Chen. MBStar: A Real-time Communication Protocol for Wireless Body Area Networks. Proceedings of the 23rd Euromicro Conference on Real-Time Systems (ECRTS), 2011. pp. 57-66, 2011.
- [133] YAN, Zhisheng; LIU, Bin. A context aware MAC protocol for medical Wireless Body Area Network. Proceedings of the 7th International Wireless Communications and Mobile Computing Conference (IWCMC), 2011. 4-8 July 2011. pp. 2133 – 2138.
- [134] ZHANG, Xiaoyu; JIANG, Hanjun; CHEN, Xinkai; ZHANG, Lingwei; WANG, Zhihua. An Energy Efficient Implementation of On-Demand MAC Protocol in Medical Wireless Body Sensor Networks. Proceedings of the IEEE International Symposium on Circuits and Systems, 2009. ISCAS 2009. pp. 3094-3097.
- [135] ZHANG, Zhi-Li; DUAN, Zhenhai; HOU, Yimei Thomas. Fundamental Trade-Offs in Aggregate Packet Scheduling. IEEE Transaction On Parallel and Distributed Systems, Vol. 16, N. 12, 2005.
- [136] ZHOU, Gang; LU, Jian; WAN, Chieh-Yih.; YARVIS, Mark D.; STANKOVIC, John.A. BodyQoS: Adaptive and radio-agnostic QoS for body sensor networks. In INFOCOM 2008: The 27th Conference on Computer Communications, pages 565--573, 2008.
- [137]